Applied Microsystems Corporation

USER'S MANUAL

8080 / 8085 Series Microprocessors

Applied Microsystems Corporation
11003-118th Place N.E.
Kirkland, WA 98033
(206) 823-9911

Toll Free Service: 1-800-426-3925

EM-188 Diagnostic Emulator*

AMC 9112 9-81

APPLIED MICROSYSTEMS™

Table of Contents

Section 1 - Introduction

Section 2 - EM-188 Components

Section 3 - Basic Operating Instructions

Section 4 - EM-188 Functions

Section 5 - RAM Overlay

Section 6 - Disassembly

Section 7 - Built-In Diagnostic Functions

Section 8 - User Implemented Code Functions

Section 9 - Supplementary Information

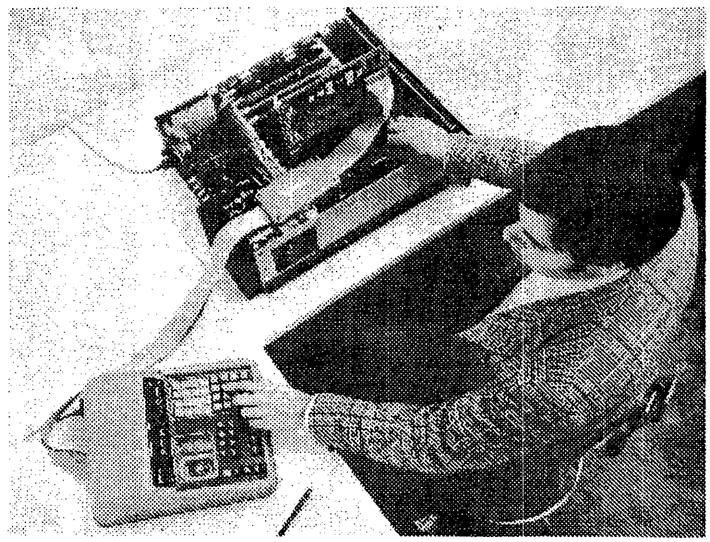
SECTION 1

INTRODUCTION

1-1 SYSTEM CONCEPT

The EM-188 Diagnostic Emulator is a microprocessor test and diagnosis instrument designed to emulate the 8085A or 8085A-1 microprocessors. An alternate cable pod and cable assembly enables the unit to emulate the 8080A, 8080A-1 and 8080A-2 microprocessors. The Diagnostic Emulator consists of an Operator's Station with Keyboard and Display Panel and an Emulator Pod and Cables for connection to the user's system. The EM-188 is fast and easy to use and includes many diagnostic capabilities for troubleshooting problems in the user's system.

Figure 1-1.1



1-2 TRANSPARENCY

The EM-188 Diagnostic Emulator is transparent to the normal operation of the target system in that emulation is in real-time, with no additional processor cycles required as a result of the emulation process. There are no target system addresses or I/O ports needed or used by the EM-188 and there are no programs or other software objects that are required to be in the target address space. As a consequence of this transparancy, the user should not experience difficulties in using the EM-188 Diagnostic Emulator with his system even if there are critical software timing constraints in his system.

1-3 WARRANTY

Applied Microsystems Corporation (AMC) warrants that the articles furnished hereunder are free from defects in material and workmanship and perform to applicable published AMC specifications for one year from date of shipment. This warranty is in lieu of any other warranty expressed or implied. In no event will AMC be liable for special or consequential damages as a result of any alleged breach of this warranty provision. The liability of AMC hereunder shall be limited to replacing or repairing, at its option, any defective units which are returned F.O.B. AMC's plant. Equipment or parts which have been subject to abuse, misuse, accident, alteration, neglect, unauthorized repair or installation are not covered by warranty. AMC shall have the right of final determination as to the existence and cause of defect. When items are repaired or replaced, the warranty shall continue in effect for the remainder of the warranty period, or for 90 days following date of shipment by AMC, whichever period is longer.

1-4 GENERAL SPECIFICATIONS

Input Power
90 to 140 Vac
60 Hz
less than 50 watts
Optional
180 to 280 Vac
50 Hz
less than 50 watts

Physical
Operator's Station

Width:	292mm (11.5 inches)
Height:	117 mm (4.6 inches)
Depth:	356 mm (14 inches)

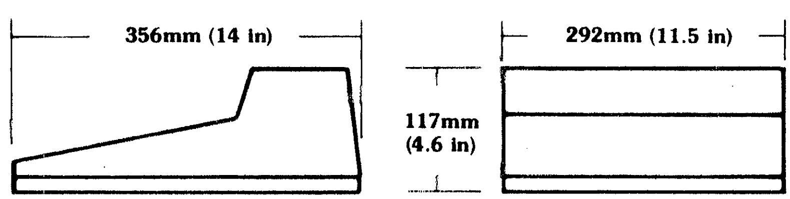
Target System Connection (Ribbon Cable)
Total Length (including Pod):
1.5M (58 inches)
Emulator Cable Pod

Width:	90mm (3.6 inches)
Depth:	33mm (1.3 inches)

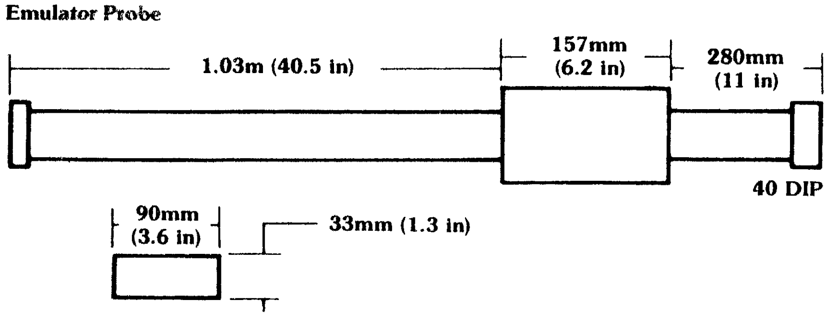
Total Weight
4.5 Kg (11 lbs); Shipping 6.3 Kg (14 lbs)

Environmental
Operating Temperature: 0°C to 40°C (32°F to 104°F)
Storage Temperature: -40°C to 70°C (-40°F to 158°F)
Humidity: 5% to 95% RH non-condensing

Operator's Station



Emulator Probe



SECTION 2

EM-188 COMPONENTS

2-1 OPERATOR'S STATION

The EM-188 Operator's Station consists of a Keyboard, Display Panel, Diagnostic EPROM Socket, Back Panel Controls and Connectors. It contains most of the system electronics, including the emulation control circuitry, Trace Memory, Breakpoint Comparators, plus control firmware with preprogrammed test routines. A RAM Overlay option may be included. See Figure 2-1.1.

2-2 EMULATOR PROBE

Two Emulator Probes are available: The EP-8085 for the 8085A and 8085A-2 microprocessors and the EP-8080 for the 8080A, 8080A-1 and 8080A-2. The Probe contains the CPU and associated circuitry and buffers. It connects to the Operator's Station via 40-inch ribbon cables and to the target system CPU socket via 11-inch ribbon cables and a 40-contact DIP connector.

2-3 KEYBOARD

The Keyboard has 32 keyswitches divided into four groupings: Processor Control, Mode Select, Subfunction Control and Data Entry.

2-4 DIAGNOSTIC EPROM SOCKET

A low insertion force EPROM socket to accept EPROMs compatible with Intel 2716 or 2732 types (single +5 power supply and Intel pinout). The user may create his own system test and diagnosis routines, program the EPROM with these routines, insert the EPROM into the EM-188 front panel socket and then execute the routines in a convenient manner from the EM-188 Keyboard. See Section 8: USER IMPLEMENTED CODE FUNCTIONS.

2-5 DISPLAY PANEL

The Display Panel consists of LED dot-matrix address and data displays and of individual LED indicators. Address and data information are displayed in hexadecimal notation. The individual indicator LEDs are divided into five groupings: Fault indicators (CLK, RESET) show loss of system clock or a continuous RESET condition; Machine Cycle indicators (Fetch, BK A, BK B, EXT, I/O, IACK, RD, WR) readout the control bus and other information acquired during target program execution; the microprocessor condition code bits (S, Z, AC, P, CY) are also displayed on these indicators; CPU Status indicators (IENA, INT, Trap, Hold, Wait, Halt, Pause) show the condition of the emulated target system CPU; Breakpoint Enable (BKPT ENA) is illuminated if the Breakpoint System is enabled.

2-6 TRACE MEMORY

The Trace Memory is a 251-word by 32-bit memory that captures information from each bus cycle of the emulated target system microprocessor. The information recorded is: the 16 address bits, 8 data bits, CPU read and write signals, the type of bus cycle (i.e., op-code Fetch, I/O, or Interrupt Acknowledge) and three possible breakpoint sources - the Breakpoint A Comparator, the Breakpoint B Comparator and the External Breakpoint input.

Figure 2-1.1. Operator's Station

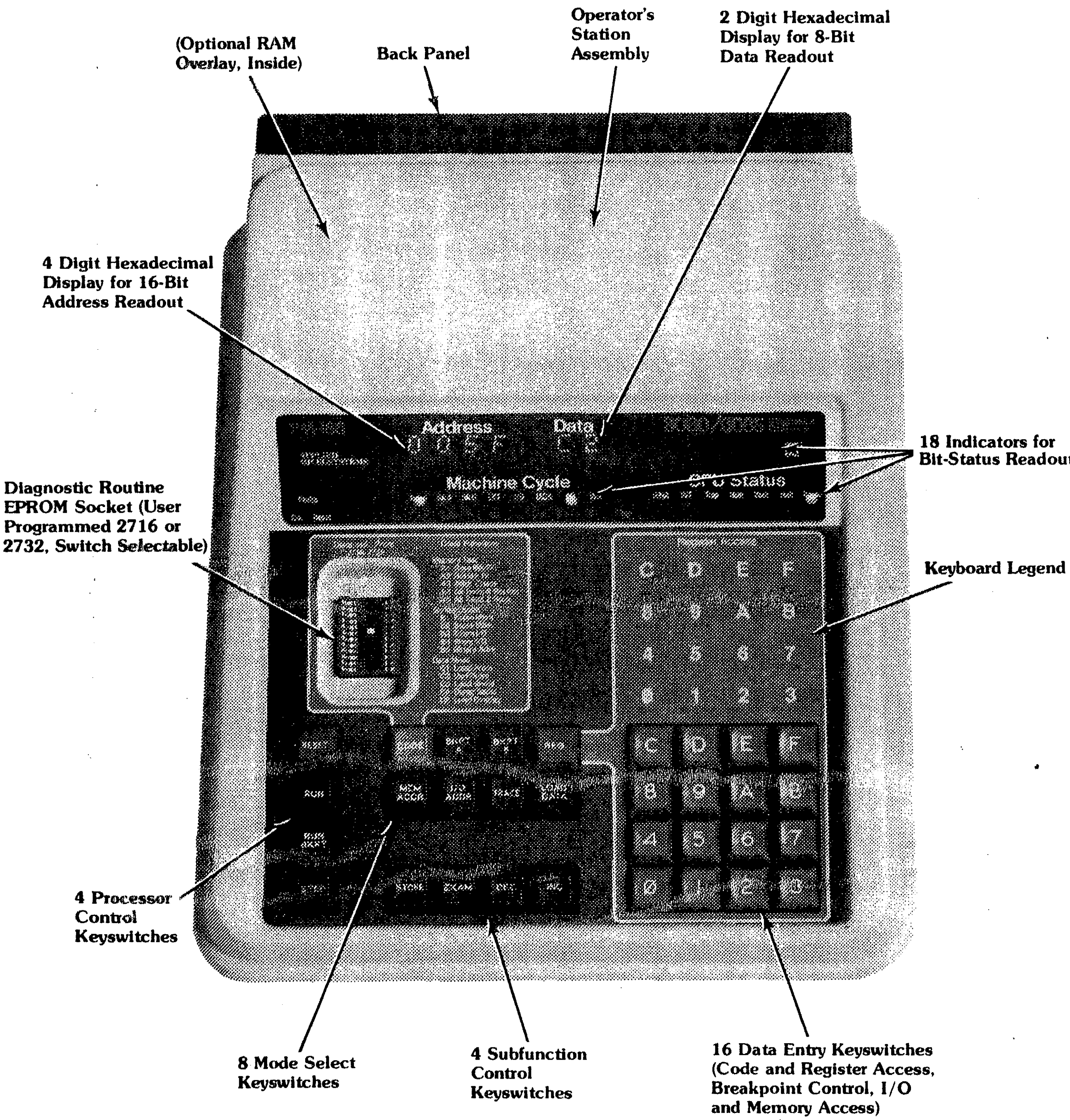
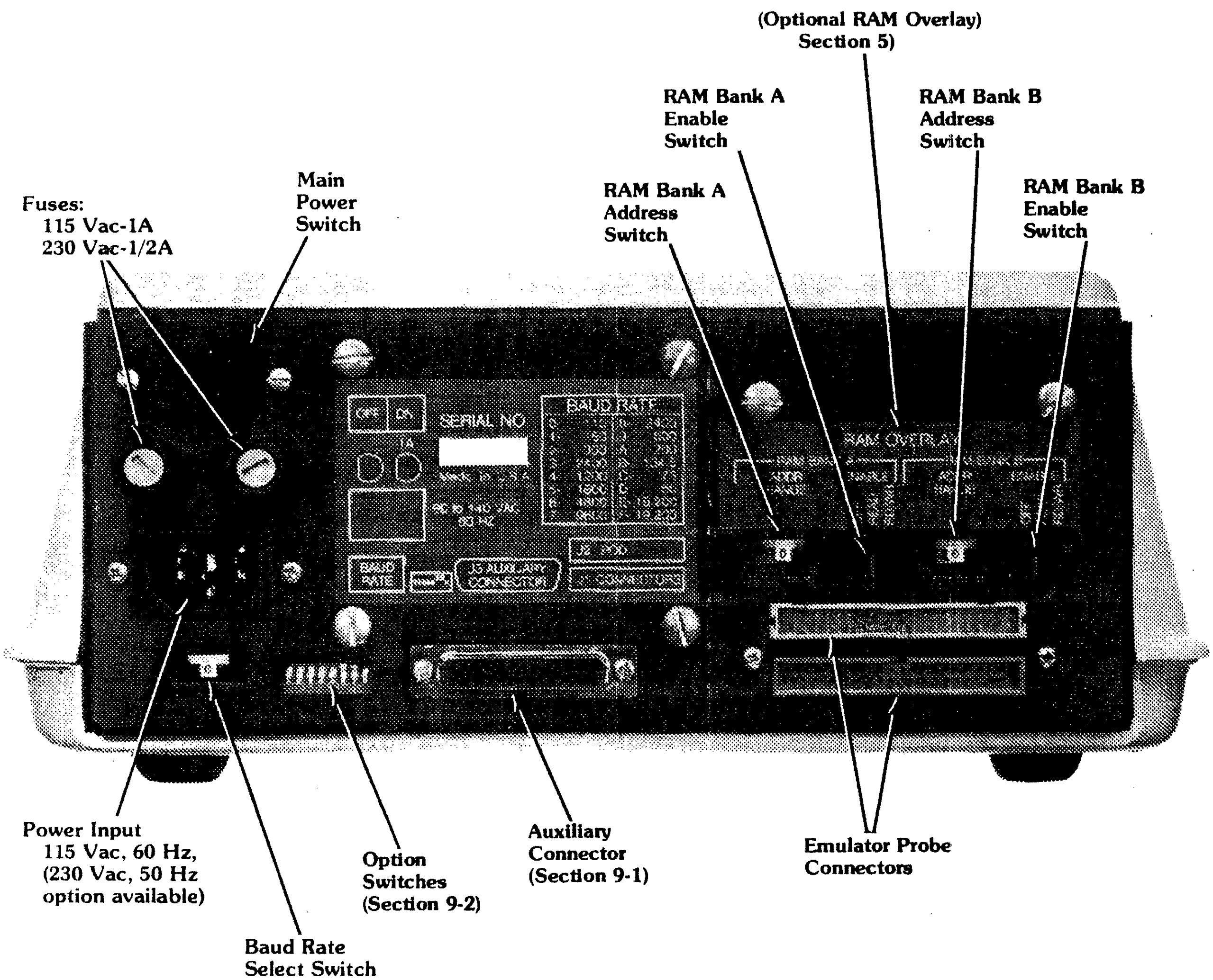


Figure 2-7.1. Back Panel



2-7 BACK PANEL CONTROLS AND CONNECTORS

The Back Panel of the EM-188 includes the controls and various connectors used to connect the Diagnostic Emulator with power, the Emulator Probe and other external equipment:

Main Power Switch
Controls the primary power to the unit.

Baud Rate Selector Switch
A 16-position switch is used to control the transmission rate of serial data flow between the Diagnostic Emulator and peripheral equipment. The baud rate selection options are visible on the Back Panel template shown in Figure 2-7.1.

Auxiliary Connector
A 25-pin, D subminiature female connector. It provides RS-232C signals and additional control signals to auxiliary equipment (i.e., signature analyzer, oscilloscope, target system, development system). See Section 9-1.

Option Switches
These switches control characteristics of the EM-188 RESET circuitry and communications interface. The normal switch positions for most users are shown in Figure 2-2. The EM-188 is shipped from the factory in this configuration. Alternative positions are discussed and illustrated in Section 9-2.

RAM Overlay Bank A and Bank B Address Switches
Two 16-position switches are used to select the address range to which the A and B blocks of enabled overlay memory responds. See Section 5, RAM Overlay.

RAM Overlay Bank A and Bank B Enable Switches
Two 3-position toggle switches control the A or B block of overlay memory. The left position (OFF) disables a memory block, effectively removing it from the system. The center position (READ) enables the memory block for read-only operations (read-only-memory simulation). The right position (RD/WR) enables a memory block for both read and write operations.

2-8 RAM OVERLAY

The EM-188 may be configured to include optional overlay memory. This feature consists of 8K bytes of 200 nsec static memory that is divided into two independent 4K byte blocks. Each block may be enabled as read-write memory, used as read- only memory or disabled. Back Panel switches are used to adjust each memory block to reside in any one of 16 address blocks in the target address space. When a memory block is enabled, it is mapped into the target address space, overlaying the user's system in the address block selected. The overlay memory may be loaded from the target system memory, front panel EPROM or external device by executing the appropriate Code Function. See Section 5, RAM Overlay.

2-9 DISASSEMBLER

The EM-188 may be configured with an optional firmware package that provides for formating and output of system information to an ASCII terminal device with RS-232C interface such as a CRT or hard copy terminal. The disassembly firmware extracts information from the EM-188 Trace Memory and emulation processor registers, formats the data for display with instruction op-codes given in standard Intel mnemonic form (LDA, LHLD, INR, JMP, etc.) and outputs the data through the serial port. See Section 6.

SECTION 3

BASIC OPERATING INSTRUCTIONS

3-1 OPERATING VOLTAGE

The EM-188 Diagnostic Emulator is normally supplied for operation from 90 to 140 Volts AC at 58 to 62 Hz line. The unit is also available to operation from 180 to 280 Volts AC at 48 to 52 Hz line if specified at time of order. The EM-188 uses a regulating transformer that also has the advantage of providing good blocking of conducted noise that may be present on the power input to the unit.

3-2 SAFETY INFORMATION

The EM-188 is supplied with a 3-wire cord with a 3-terminal polarized plug for connection to the power source and protective ground. The ground terminal of the plug is connected to the metal chassis parts of the instrument. Electric-shock protection is provided if the plug is connected to a mating outlet with a protective ground contact that is properly grounded.

The internal (logic) ground of the EM-188 is not connected to the protective ground, but floats to the same potential as the equipment to which the unit is connected. The user is cautioned that it is conceivable that grounding conflicts could occur if the EM-188 is connected to two different items of equipment with differing ground potentials such as the target equipment and a RS-232C terminal.

3-3 CONNECTION TO TARGET EQUIPMENT

Connect the EM-188 Diagnostic Emulator to the target system by removing the microprocessor from its socket in the user's system and plugging in the 40-pin plug of the EM-188. (See Figure 3-3.1.)

CAUTION: NOTE CORRECT PIN 1 ORIENTATION

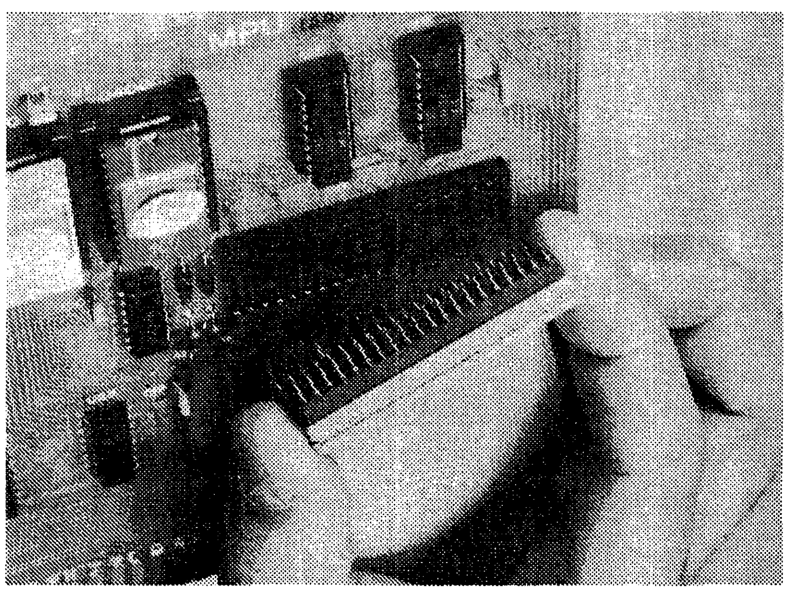


Figure 3-3.1. Installing 40-pin plug.

BASIC OPERATING INSTRUCTIONS

Apply power to the EM-188 and the target system after the unit is properly connected to the target circuitry. Once power is applied to the Diagnostic Emulator and the clock begins operating, it performs a Power-on-reset operation during which the following functions are performed:

1. Reset CPU.
2. Clear Trace Memory and CPU Registers.
3. Clear the program starting address to zero (the default starting address) and display the address.
4. The Diagnostic Emulator awaits further input from the operator.

After the EM-188 is connected to the target system and power is applied to both the EM-188 and the target system, perform the following checks to verify that the unit is operating correctly:

1. The Clock Fault and Reset Fault indicators are not illuminated. This means that the 8085 clock oscillator is operating (or if an 8080, that clock signals are present from the target system) and that there is not a continuous RESET signal from the target system.
2. The PAUSE indicator should be illuminated. This indicates that no target program is executing and that the EM-188 is awaiting operator commands.
3. At power on time, the ADDRESS Display should indicate 0000. If it does, the EM-188 internal control program is operating.

If all is well, proceed to operate the Diagnostic Emulator as appropriate. See Section 4 for details of EM-188 functions.

SECTION 4

EM-188 FUNCTIONS

A basic function of the EM-188 is to emulate the target system microprocessor. Effectively, the Diagnostic Emulator is a pin-compatible functional replacement for the microprocessor in the target system. The unit is designed to meet the timing specifications of the emulated processor and to minimize the increase in electrical loading of the user's system.

The EM-188 is always in one of two modes: RUN or PAUSE. If in the RUN mode, the EM-188 is emulating the target system microprocessor and executing the target system program at full system speed. The Trace Memory will be active (unless inhibited by external control) and all bus cycles of the emulated microprocessor are recorded for possible later display. In the PAUSE mode, emulation of the target system microprocessor is suspended and the operator is able to perform other functions such as manually examining or altering memory locations, I/O ports or internal registers of the emulated microprocessor; the operator may also review the history of the target program execution from the Trace Memory or execute one of the Code Function routines.

4-1 EXECUTION CONTROL

The operator controls the EM-188 primarily through the Operator's Station Keyboard. Keyswitch groupings are designed for easy understanding and convenient use. The EM-188 display provides the operator with information about program execution, CPU status and EM-188 conditions. Table 4-1.1 defines the Display Panel indicators.

4-1.0 RESET Keyswitch

The red RESET Keyswitch always resets the microprocessor and initializes the EM- 188 in the PAUSE mode. At this time the Address Display shows the program starting address. The program starting address may be changed at this time by entering digits with the hexadecimal keyswitches, or the current program starting address may be used.

The option switches accessible at the Back Panel of the EM-188 may be used to set up one of several options concerning the RESET circuitry of the EM-188 and the target system. See Section 9-2.

4-1.1 RUN Keyswitch

Pressing the RUN Keyswitch causes the EM-188 to execute the target program beginning at the preset address or continuing from the last instruction executed. Execution is at full system speed with no extra wait states beyond those commanded by the target system. The activity of the executing program is recorded continuously in the Trace Memory. It is also possible to obtain a general view of the program activity by watching the displays. For example, it is possible to tell if the program is in a tight loop or ranging widely in the program address space by observing changes in the Address Display.

4-1.2 RUN BKPT Keyswitch

This Keyswitch starts the EM-188 running the target program in real time, as does RUN, except the breakpoint-stop circuitry is enabled. If a breakpoint is detected, the EM-188 will pause before executing the next instruction, and the display will show the cycle where the breakpoint was detected. Pressing RUN BKPT again will cause execution to resume until the breakpoint is again detected. The breakpoint- stop circuitry may be disabled during program execution by pressing RUN.

4-1.3 STEP Keyswitch

Pressing the STEP Keyswitch while the program is running causes the program execution to stop. The displays at this point show the operation code fetch cycle of the last instruction executed, with the address, op-code (data) and control signals visible. When the Diagnostic Emulator stops executing the target program, the following events take place:

1. The processor stops executing the target program.
2. The processor registers are saved in internal scratch pad memory and are accessible for display or alteration.

The operator, in effect, freezes the target program execution at the point reached when STEP was depressed. The operator then has several options:

1. Continue executing the program at full speed by pressing RUN.
2. Continue executing the program one instruction at a time by pressing STEP for each additional instruction execution.
3. Examine or change the contents of any of the processor registers.
4. Examine any memory or I/O address, and if the location is writable, store new data in it.
5. Review the last 251 bus cycles performed by the processor by decrementing through the Trace Memory.

The state of the target program is not changed by any of these operations (except as purposely altered by the operator) and program execution may be continued from the point where it stopped.

The program may be executed one instruction at a time by pressing STEP once for each instruction. If STEP is pressed and held down, the Diagnostic Emulator begins stepping at about seven instructions per second. The step rate then accelerates gradually from 7 steps per second to about 50 steps per second. Execution stops again if the keyswitch is released.

Table 4-1.1. Display Panel Indicators

FAULT GROUP

	ILLUMINATES IF:
CLK	Target system clock not operating. Target system clock is low in frequency. EM-188 not connected to target system.
RESET	Processor and Diagnostic Emulator held in Reset by a low on the RESET IN terminal of the microprocessor socket.

MACHINE CYCLE GROUP

	ILLUMINATES IF:
FETCH	Displayed machine cycle is an op-code fetch cycle.
BK A	Breakpoint A. Conditions set up for an output from Breakpoint A Comparator were satisfied during the displayed machine cycle.
BK B	Breakpoint B. Conditions set up for an output from the Breakpoint B Comparator were satisfied during the displayed machine cycle.
EXT	External. External Breakpoint input low (active) during displayed machine cycle.
I/O	Input/Output. Machine cycle being displayed is a data transfer from an input port address or to an output port address.
IACK	Interrupt Acknowledge. Machine cycle being displayed is an interrupt acknowledge cycle.
RD	Read. Machine cycle being displayed is read from memory or read from I/O cycle.
WR	Write. Machine cycle being displayed is a write to memory or write to I/O cycle.

FLAGS

	ILLUMINATES IF:
S	Sign bit is true.
Z	Zero bit is true.
AC	Auxiliary Carry bit is true.
P	Parity bit is true.
CY	Carry bit is true.

CPU STATUS

	ILLUMINATES IF:
IENA	Emulated processor is ready to respond to an interrupt. (Interrupts enabled)
INT	Interrupt. One of the following inputs to 8085 is active: INTR (PIN 10); RST 5.5 (PIN 9); RST 6.5 (PIN 8); RST 7.5 (PIN 7). For the 8080, the INT (PIN 14) input is active.
Trap	TRAP (PIN 6) input to 8085 is active. Not applicable to 8080.
Hold	HOLD input of the 8085 or 8080 active.
Wait	Wait. READY input to 8085 (PIN 35) or 8080 (PIN 23) is not active. CPU is waiting. If the CPU is working with a system that requires some wait states, the indicator may be more or less dimly lit.
Halt	Processor has executed a HALT instruction and has entered the HALT state.
Pause	Real-time emulation of the target program is suspended and the Diagnostic Emulator is awaiting another command.
BKPT ENA	Breakpoint Enable. Illuminates if the RUN BKPT keyswitch is depressed.

4-1.4 Breakpoint System

The Diagnostic Emulator incorporates a Regional/Relational breakpoint generation system to enable the user to monitor the operation of his program and to stop execution of his program when desired. The EM-188 contains two independent address comparators. Each of these comparators continuously monitors the 16-bit address bus of the microprocessor. In addition, each comparator may be qualified to respond to read cycles only, to write cycles only, or to both read and write cycles. The comparators may also be configured to respond to memory addresses or to I/O port addresses.

It is also possible to configure the breakpoint system so that a specified relationship must hold between the A and B breakpoint comparators before PAUSE occurs. The relationships that may be specified are the following:

1.	A then B.	Break if condition A is found followed some time later by condition B.
2.	A◄►B	Break if any address in the range from A to B (inclusive) is found.
3.	◄A▬B►	Break if any address outside of the range from A to B is found. (Including addresses A and B.)

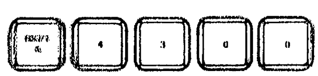
Table 4-1.2. Breakpoint Qualifiers

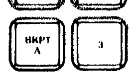
0 - Disable	5 - I/O Read
1 - Memory Read	6 - I/O Write
2 - Memory Write	7 - I/O Read/Write
3 - Memory Read/Write	8 - Range A to B
4 - A then B	9 - Range outside A to B

The various breakpoint qualifiers and relationships are set up by simple keystroke sequences. Some examples of these sequences follow.

EXAMPLE:
Set up breakpoint comparator A to respond to read or write cycles at address 4300_{16}; disable comparator B.

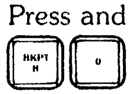
KEYSTROKE SEQUENCE:


Set breakpoint address.



Set qualifier 3 (memory R/W).

Press and hold down BKPT A Key while qualifier is entered.

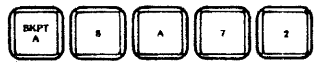


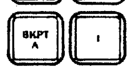
Set qualifier 0 (Disable).

Press and hold down BKPT B Key while qualifier is entered.

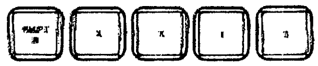
EXAMPLE:
Set up breakpoint comparator A to respond to read cycles only at memory address $8A72_{16}$, and breakpoint comparator B to respond to write cycles to I/O port 13_{16}.

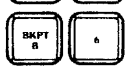
KEYSTROKE SEQUENCE:


Set A breakpoint address.


Set A qualifier to 1 (MEMORY read).

Press and hold down BKPT A Key while qualifier is entered.


Set B breakpoint address.

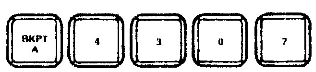

Set B qualifier to 6 (I/O WRITE).

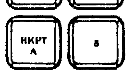
Press and hold down BKPT B Key while qualifier is entered.

When the breakpoint circuitry is set up as desired, program execution may be started using RUN BKPT. The function RUN BKPT is the same as the function of RUN except that when the breakpoint condition occurs, program execution stops.

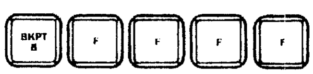
EXAMPLE:
Set up breakpoint range from 4307_{16} to $FFFF_{16}$.

KEYSTROKE SEQUENCE:


Set A to range beginning (4307_{16}).


Set A qualifier to 8 (Range A to B).

Press and hold down BKPT A Key while qualifer is entered.


Set B to range end ($FFFF_{16}$).

Note that the qualifier for B is automatically set to eight when the range qualifier was selected for A, since both breakpoint qualifers are affected when selecting A then B or RANGE qualifiers.

4-1.5 TRACE MEMORY

One of the most useful EM-188 features is its 251 bus cycle Trace Memory. The Trace Memory is organized as a ring buffer that records all target program activity. It operates in both real-time and single-step modes, and its contents remain in the correct sequence, regardless of whether the user operates the program wholly or partly in either of these two modes.

To review the Trace Memory contents, the EM-188 must be paused. The PAUSE mode is entered automatically when the program encounters a breakpoint, or it can be entered manually by depressing STEP. When the program enters PAUSE, the Display shows the fetch cycle address and data for the last instruction recorded.

When a breakpoint event triggers PAUSE, the Display shows the cycle where the breakpoint was detected, and the user can easily review the program activities leading up to the event. Depressing DEC allows the user to examine the last 251 bus cycles of program activity prior to the breakpoint. Depressing INC allows the user to review forward up to the last cycle traced. Depressing STEP advances the target program past the breakpoint event, one instruction at a time. Depressing TRACE allows the user to return to Trace Memory again after selecting another mode (i.e., MEM ADDR, I/O ADDR, etc.) and return the original program event or bus cycle to the display. The TRACE Key has no effect unless the program is already in PAUSE. STEP actually causes the emulator to execute another program instruction and this instruction is entered into the Trace Memory like all others.

The 8085/8080 machine instructions may have one or several bus cycles per instruction. The following two examples illustrate displayed Trace Memory contents, first after executing a simple instruction and then after a more complex one.

EXAMPLE 1 MOV B, C

Cycle	Addr	Data	Fetch	RD	WR
1	4000	41	•	•	

Single bus cycle instruction: Move contents of C register to B register. Assume the instruction location is address 4000_{16} in the target memory. The Trace Memory records a bus cycle with the address of 4000_{16}, data of 41_{16} and control bits indicating that it is a fetch operation and a read cycle.

EXAMPLE 2 SHLD 7055H

Cycle	Addr	Data	Fetch	RD	WR
1	4000	22	•	•	
2	4001	55		•	
3	4002	70		•	
4	7055	34			•
5	7056	12			•

Five bus cycle instruction: Cycle one fetches op-code 22 of the SHLD instruction located at address 4000_{16}. Cycles two and three read low order and high order bytes (55_{16} and 70_{16}) of the 16-bit address located at 4001_{16} and 4002_{16}. Cycles four and five write the contents of the HL register pair (34_{16} and 12_{16}). The Trace Memory records all five bus cycles of the instruction. The address location, program data and cycle type are shown on the Display Panel for each bus cycle of the instruction. If the EM-188 had entered PAUSE and displayed Cycle 1 (the OP-CODE fetch), then the INC Key would be used to advance through the Trace Memory and observe the subsequent bus cycles.

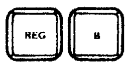
4-2 EXAMINATION AND ALTERATION OF CPU REGISTERS

The 8085 and 8080 register contents may be examined by the operator, and if desired, overwritten with new data.

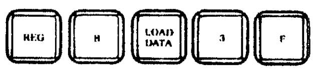
Register data is accessed for display by using the blue REG Keyswitch, followed by one of the hexadecimal keyswitches. This designates which register should be displayed. Table 4-2.1 shows the registers selected by the various keyswitches. Note that 4 through 7 do not correspond to actual 8085 or 8080 registers. These keyswitches are used to set up parameters for the Built-In test routines or User Code Functions. These Code Functions are described in later sections. (See Section 7 and Section 8).

Examples of readout and alteration of CPU registers:

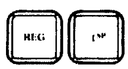
EXAMPLE:



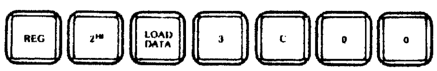
B register contents displayed on data display.



B register is accessed and then overwritten with data $3F_{16}$.



Stack Pointer is accessed and displayed on 16-bit address display.



HL Register Pair is accessed, and then the contents are overwritten with $3C00_{16}$.

Table 4-2.1. Keyboard Designators

KEY	REGISTER	DESCRIPTION
0	PC	Program Counter
1	SP	Stack Pointer
2	HL	HL Register Pair
3	DE	DE Register Pair
4	* BEG	BEGIN Address (for programmed tests)
5	* END	END Address (for programmed tests)
6	* ADDR	Address (programmed test parameter)
7	* DATA	Data (programmed test parameter)
8	** RIM	Current Status of Interrupt Mask Register
9	** SIM	Data to load to Interrupt Mask Register
A	A	Accumulator
B	B	B Register
C	C	C Register
D	D	D Register
E	E	E Register
F	F	Condition Flags

* *Not an actual 8080 or 8085 register.*
** *Not used for 8080 emulation.*

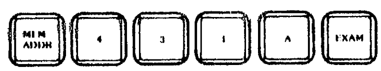
EXAMPLE:

SIM Register is accessed and then overwritten with data $C0_{16}$.

4-3 EXAMINATION AND ALTERATION OF MEMORY LOCATIONS

Any memory location accessible to the emulated microprocessor may be accessed and displayed by the EM-188. If desired, new data may be written to the location.

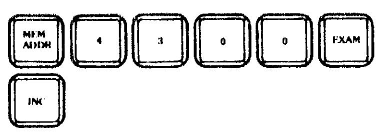
EXAMPLE:



Address $431A_{16}$ is entered, and when EXAM is pressed, the EM-188 will read from address $431A_{16}$ and display the data obtained.

If the operator wishes to review a group of sequential memory locations, this may be done by entering the initial address and examining that location as above; then examine successive locations by depressing INC.

EXAMPLE:



Examine data at 4300_{16}.

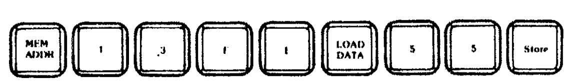
Examine data at 4301_{16}

. . . etc.

Examine data at successive locations, etc.

A memory location may be altered by entering an address, as shown above, then entering data using LOAD DATA and finally storing the data to the selected memory address using STORE.

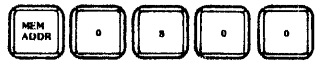
EXAMPLE:



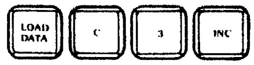
The above sequence writes the data 55_{16} to memory address $13FE_{16}$ in the target system.

Sequential locations may be quickly altered by incrementing the address after each data entry operation. For example, the following keystroke sequence enters a short program fragment into memory:

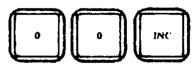
EXAMPLE:



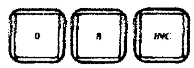
Enter initial address 0800_{16}.



Enter data $C3_{16}$ then store the data to 0800_{16} and increment to 0801_{16}.



Enter data 00_{16} then store the data to 0801_{16} and increment to 0802_{16}.



Enter data 08_{16} and store to 0802_{16}; increment to 0803_{16}, etc.

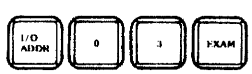
The EM-188 does not require redundant keystrokes. The unit assumes that if the operator has entered new data while a particular memory address is accessed, then the operator wants to store that data before going to the next address.

In all of the above examples in which INC was used, DEC (decrement) could also have been used.

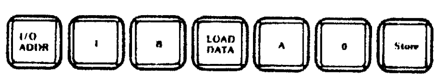
4-4 EXAMINATION AND ALTERATION OF I/O PORTS

Input/Output ports of the 8085 or 8080 may be accessed and displayed in a parallel manner to that described for memory addresses, with two differences. The first is that the I/O ports respond to an eight-bit address and consequently only eight-bit addresses need be entered. The second difference is that the INC and DEC keyswitches do not perform an automatic read of the next (or previous) I/O port address. The intent of this characteristic is to help the operator avoid unintended reading of an I/O port, since this sometimes results in a change of state of complex I/O devices. For example, a communications interface circuit such as the Intel 8251 will clear the DATA PRESENT status bit when the receive data register is read. The following is a keystroke sequence that may be used to examine data at an input port:

EXAMPLE:



Read and display data at Input Port 3.



Data $A0_{16}$ is written to Output Port $1B_{16}$. (No READ cycle was performed at Port 1B.)

SECTION 5

RAM OVERLAY

5-1 OVERVIEW

The EM-188 may be configured with an optional RAM Overlay feature. The RAM Overlay consists of a circuit board with 8K bytes of 200 nSEC static memory along with appropriate addressing and control logic.

The 8K bytes of memory is divided into two independent 4K byte blocks. Each block has independent addressing and control circuitry and may be enabled as read- write memory, as read-only memory or disabled. Controls for the two memory blocks are located on the back panel of the Operator's Station.

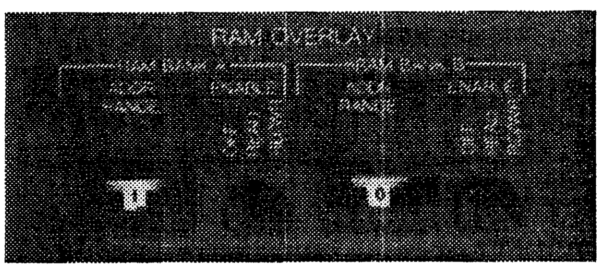
If the RAM Overlay is set up to respond to a range of addresses, say 0000_{16} to $0FFF_{16}$, then target system memory in the same address range becomes inaccessible to the emulation processor. The memory block has "overlayed" the corresponding target system addresses. (See, however, the description of Code Function C5 for an exception to this characteristic of the EM-188.)

The contents of the RAM Overlay is retained as long as power is applied to the EM- 188. It is possible to load the RAM Overlay with data, then switch the enable switches to the OFF Position, then at some later time, switch the enable switches to READ or RD/WR and use the data.

5-2 CONTROLS

Each 4K byte block of memory has an associated Address Range switch and Enable switch. See Figure 5-2.1.

Figure 5-2.1. RAM Overlay Controls.



The Enable switches are three-position toggle switches that place the memory bank into one of three conditions:

1. Off

The memory bank is disabled and is effectively removed from the system.

2. Read

The memory bank is placed in a Read-Only configuration so that from the point of view of the target system the memory bank behaves like Read-Only-Memory. In this mode, it is not possible for the target system program to alter the contents of the memory. Note, however, that the EM-188 is still able to write to the memory bank from the Keyboard or from a Code Function routine such as Code Function C3 (download).

3. Read/Write

The memory bank is placed in a Read/Write configuration. Both the target system and the EM-188 are able to read the memory and write new information to it.

If a memory bank is disabled (toggle switch in the OFF position), the memory will nevertheless continue to retain data. The data will reappear in the target address space whenever the memory is again enabled.

The Address Range switches are 16-position rotary switches used to select the address range where the 4K memory blocks will reside in the target address space. Each of the 4K memory blocks can be moved to any of 16 positions, beginning at a 4K boundary. See Table 5-2.1.

Table 5-2.1. Memory Block Address—A and B

SWITCH POSITION	MEMORY BLOCK ADDRESS	SWITCH POSITION	MEMORY BLOCK ADDRESS
0	0000_{16} - $0FFF_{16}$	8	8000_{16} - $8FFF_{16}$
1	1000_{16} - $1FFF_{16}$	9	9000_{16} - $9FFF_{16}$
2	2000_{16} - $2FFF_{16}$	A	$A000_{16}$ - $AFFF_{16}$
3	3000_{16} - $3FFF_{16}$	B	$B000_{16}$ - $BFFF_{16}$
4	4000_{16} - $4FFF_{16}$	C	$C000_{16}$ - $CFFF_{16}$
5	5000_{16} - $5FFF_{16}$	D	$D000_{16}$ - $DFFF_{16}$
6	6000_{16} - $6FFF_{16}$	E	$E000_{16}$ - $EFFF_{16}$
7	7000_{16} - $7FFF_{16}$	F	$F000_{16}$ - $FFFF_{16}$

SECTION 6

DISASSEMBLY

6-1 OVERVIEW

The EM-188 Diagnostic Emulator may be configured with an enhanced firmware package that includes a disassembler. The disassembler firmware gives the EM-188 the ability to output the contents of the Trace Memory and emulation processor registers to the serial port; in this way a readable and attractive display may be created on a CRT or hardcopy terminal.

The disassembly firmware is disabled when the EM-188 is first powered up and must be enabled before use. The following procedure will make the EM-188 ready to operate with an ASCII terminal and disassembly firmware:

6-1.1 OPERATION PREPARATION PROCEDURES

1. Connect the terminal to the EM-188 using an appropriate cable. The minimum circuits that must be connected are:

   Pin 1 - Frame Ground

   Pin 2 - Serial Data Out

   Pin 7 - Ground

Some RS-232 terminals may also require the following connection:

   Pin 20 - Data Terminal Ready

   Take care that Pins 10, 11, 12, 13, 22, 23, 24 and 25 are not connected to incompatible circuits. See 9-1, Auxiliary Connector.

2. Set the Baud Rate Selector switches of the EM-188 and the terminal to compatible settings.
3. Check the setting of Option Switch 3. If Option Switch 3 is open (up), then the EM-188 will not output serial data unless the Clear-to-Send signal (Pin 5) is high. If the Clear-to-Send signal is not important in your application, set Option Switch 3 to the CLOSED (down) position and the EM-188 will output data on command regardless of the state of Pin 5.
4. Enable the disassembly firmware by executing the Code Function E1. The EM-188 is now ready to output lines of up to 72 characters to the terminal device. Since 72 characters is not a long enough line to include the interrupt mask register display, this register will not appear.
5. If the ASCII terminal is capable of displaying lines of 80 characters or longer, enable the display of 80 character lines by executing Code Function E3.

The disassembly firmware may be turned off by executing Code Function E0. The 80-character line may be turned off by executing Code Function E2.

Operate the EM-188 in the normal manner. Any time that the EM-188 transfers from RUN to PAUSE, the disassembly firmware will format and dump a part of the contents of the Trace Memory to the terminal; normally 24 lines of output are produced. The last line output represents the last instruction executed and the firmware will then output the register display.

If the EM-188 is operated in single-step mode, the firmware will be able to output the register display after every instruction.

6-2 FORMAT DEFINITION

Figure 6-2.1 shows some lines from a printer connected to our EM-188. The program was executed in single-step mode; as a consequence, a register display is obtained after each instruction execution. The various fields of the disassembly presentation are identified in the figure. All numbers that are output by the disassembler are in hexadecimal representation. Additional information about the fields of the display follows:

Address

The address of the op-code byte of the instruction.

Op-Code

The operation code of the instruction.

Operand

The operand bytes of the instruction (if any).

Op-Code Mnemonic

The operation code of the instruction given in mnemonic form.

Operand

The operand field of the instruction in symbolic format, except that addresses and constants are given as hexadecimal numbers.

Data Transfer

Any data transfer operations that occur as a consequence of the instruction are shown here. The most common formats are:

AAAA > DD

or AAAA < DD

The first format means that the processor wrote data 'DD' to address 'AAAA'. The second format means that the processor read data 'DD' from address 'AAAA'. The other formats are associated with I/O instructions and take the form:

—PP > DD

or —PP < DD

The first of these formats means that the processor wrote data 'DD' to output port 'PP'. The second format means that the processor read data 'DD' from input port 'PP'.

Data Transfer

Some instructions transfer more than one byte of data. The second byte of a data transfer will be shown in this field. If there are more than two bytes transferred, the additional bytes are shown in fields 6 and 7 on the following line. See, for example, the XTHL instruction which reads two bytes from the top of stack and then writes two other bytes to the top of stack. All of these transfers are easily seen from the display.

Breakpoint

If a breakpoint occurred during the execution of the instruction on this line, it will be identified in this column. If the Breakpoint A conditions were satisfied, an 'A' will appear in this column. If the Breakpoint B conditions were satisfied, a 'B' will appear in this column. If both the A and B conditions are satisfied during the execution of the same instruction, the 'B' will take precedence and be displayed.

Flag Register

The CPU flag register (condition code register) is shown in this field. Each of the five characters in this field represent one of the condition code bits as follows:

First - 'S' if sign bit is true.
Second - 'Z' if zero bit is true.
Third - 'A' if auxiliary carry bit is true.
Fourth - 'P' if parity bit is true.
Fifth - 'C' if carry bit is true.

If any of the condition code bits are not true, the letter is replaced by a period.

Accumulator

The content of the accumulator after the execution of the instruction.

BC Register Pair

The content of the BC register pair following the execution of the instruction.

DE Register Pair

The content of the DE register pair following the execution of the instruction.

HL Register Pair

The content of the HL register pair following the execution of the instruction.

Stack Pointer

The content of the stack pointer following the execution of the instruction.

Interrupt Mask

The content of the interrupt mask register following the execution of the instruction.

Figure 6-2.1 Disassembly format.

ADDRESS	OP-CODE	OPERAND	OP-CODE MNEMONIC	OPERAND	DATA TRANSFER	DATA TRANSFER	BREAKPOINT	FLAG REGISTER	ACCUMULATOR	BC REGISTER PAIR	DE REGISTER PAIR	HL REGISTER PAIR	STACK POINTER	INTERRUPT MASK REGISTER
0053	3A	0034	LDA	3400	3400<FF			.Z.P.	FF	0000	62DD	0000	30EB	
0056	E6	40	ANI	40				..A..	40	0000	62DD	0000	30EB	
0058	C2	8C00	JNZ	008C				..A..	40	0000	62DD	0000	30EB	
008C	21	9F30	LXI	H,309F				..A..	40	0000	62DD	309F	30EB	
008F	7E		MOV	A,M	309F<FF			..A..	FF	0000	62DD	309F	30EB	
0090	A7		ANA	A				S.AP.	FF	0000	62DD	309F	30EB	
0091	F2	9F00	JP	009F				S.AP.	FF	0000	62DD	309F	30EB	
0094	36	40	MVI	M,40	309F>40			S.AP.	FF	0000	62DD	309F	30FB	
0096	DB	36	IN	36	--36<17			S.AP.	17	0000	62DD	309F	30EB	
0098	32	8C30	STA	308C	308C>17			S.AP.	17	0000	62DD	309F	30EB	
009B	3D		DCR	A				..A..	16	0000	62DD	309F	30EB	
009C	32	8A30	STA	308A	308A>16			..A..	16	0000	62DD	309F	30EB	
009F	16	60	MVI	D,60				..A..	16	0000	60DD	309F	30EB	
00A1	3A	9530	LDA	3095	3095<03			..A..	03	0000	60DD	309F	30EB	
00A4	CD	E402	CALL	02E4	30EA>00	30E9>A7		..A..	03	0000	60DD	309F	30E9	
02E4	47		MOV	B,A				..A..	03	0300	60DD	309F	30E9	
02E5	FE	08	CPI	08				S...C	03	0300	60DD	309F	30E9	
02E7	0E	05	MVI	C,05				S...C	03	0305	60DD	309F	30E9	

SECTION 7

BUILT-IN DIAGNOSTIC FUNCTIONS

The Diagnostic Emulator contains Built-In test functions and utility routines designed to be convenient and useful for testing systems and verifying their proper operation. These test and utility routines have been named Code Functions because they are accessed by depressing the CODE keyswitch, followed by hexadecimal digits designating the routine desired. Table 7-1 lists all of the Code Functions programmed into the Diagnostic Emulator.

7-1 GROUP A: MEMORY TESTS

CODE A1 - 00/FF DATA TEST

The Code Function A1 memory test routine quickly determines whether all locations within a specified range can be set to 00_{16} and FF_{16}. The range tested is from the address specified by the BEG (begin) register through the address specified by the END register. The routine operates by setting the first location of the range to 00_{16} and reading the location to see if a 00_{16} is returned. Then the routine stores an FF_{16} to this location and reads the location to see if an FF_{16} is returned. Finally, the routine increments the address and tests the next location in the range. When all locations in the range have been tested, the EM-188 will emit a short beep and the test will repeat. During the execution of this test, the address and data activity are visible on the displays and stored in the Trace Memory.

If a memory error is encountered, the routine emits 3 beeps and displays the address of the failure and the erroneous data read. At this time the operator has three options:

1. Depress EXAM to display the data the routine expected to read from the memory. Relese the keyswitch to again display the bad data.
2. Depress INC to continue testing at the next address in the range. If additional problems are found, the program will stop again and any of the options listed may be taken.
3. Exit the test routine by using any of the mode keys (MEM, I/O, REG, RUN, etc.) or RESET.

After testing all locations in the specified range, the EM-188 emits one short beep and repeats the test. The RESET keyswitch is used to terminate the test at any time.

Table 7-1. Code Functions

GROUP A: MEMORY TESTS

A1	RAM TEST	(00/FF)	Repeating Tests
A2	RAM TEST	(Rotating 1s)	
A3	RAM TEST	(Addresses)	
A4	ALL RAM TESTS		
A5	ALL RAM TESTS		One Pass and Stop
A6	RAM TEST	(00/FF)	
A7	RAM TEST	(Rotating 1's)	
A8	RAM TEST	(Addresses)	

GROUP B: OSCILLOSCOPE LOOPS

B1	Repetitive Memory Read
B2	Repetitive Memory Write
B3	Repetitive I/O Read
B4	Repetitive I/O Write
B5	Continuous Address Increment
B6	Repetitive Memory Write (Data / $\overline{\text{Data}}$)
B7	Repetitive I/O Write (Data / $\overline{\text{Data}}$)
B8	Toggle SOD Line (8085 only)

GROUP C: MEMORY LOAD AND DUMP

C1	Load Target from Front Panel PROM
C2	Verify Target with Front Panel PROM
C3	Load Target from Serial Link (DOWNLOAD)
C4	Dump Target to Serial Link (UPLOAD)
C5	Load RAM Overlay from Target
C6	Verify RAM Overlay with Target
C7	Verify Target Against Serial Link
C8	Fill Target with Specified Data
C9	Verify Target with Specified Data

GROUP D: MISCELLANEOUS

D0	Clear Interrupt Enable Flip-Flop
D1	Set Interrupt Enable Flip-Flop
D2	Display Clock Frequency
D3	Display PROM/ROM Signature
D4	Output 50 Nulls from Serial Link
D5	Call User Routine in Internal RAM at 3000_{16}
D6	Call User Routine in Internal RAM at 3003_{16}
D7	Clear Trace Memory
D8	Call Remote Control Software
D9	Halt CPU (for changing Front Panel PROM)

GROUP E: CHANGE DEFAULT PARAMETERS

E0	Disable Disassembly (Default)
E1	Enable Disassembly
E2	Output 72 Character Lines (Default)
E3	Output 80 Character Lines

GROUP F:

F	Set "Introspection" Mode (for self testing)

CODE A2 - ROTATE 1s

Code Function A2 memory test routine performs a test on all data bits in the range specified. The range tested is from the address contained in the BEG register through the address contained in the END register. The routine starts with the first location in the range and tests the location by writing and checking a bit, one bit at a time, in all of the positions of the word under test. The routine writes and checks by writing and reading the following data patterns:

Binary Pattern	Hexadecimal
0 0 0 0 0 0 0 1	01_{16}
0 0 0 0 0 0 1 0	02_{16}
0 0 0 0 0 1 0 0	04_{16}
0 0 0 0 1 0 0 0	08_{16}
0 0 0 1 0 0 0 0	10_{16}
0 0 1 0 0 0 0 0	20_{16}
0 1 0 0 0 0 0 0	40_{16}
1 0 0 0 0 0 0 0	80_{16}

After a location has been tested it is known that all bit positions in the location may be set and cleared independently of each other. The program then increments to the next sequential address in the range and proceeds to test in the same manner. If an error is detected, the test stops, the EM-188 emits the three beeps that signify an error, and the Display Panel shows the defective memory address and the bad data. At this point the operator has three options:

1. Depress EXAM to display the data the diagnostic routine expected to read (good data). Release EXAM to return the bad data to the display.
2. Depress INC to continue testing. If additional problems are found, the test stops and any of the options listed may be taken again.
3. To terminate the test, depress RESET, RUN, or any of the mode select keyswitches.

After testing all locations in the specified range, the EM-188 emits one short beep and repeats the test. The RESET keyswitch is used to terminate the test at any time.

CODE A3 - ADDRESS TEST

Code Function A3 memory test determines whether an address decoding failure exists in the memory system under test. It tests the memory range from the address contained in the BEG register to the address contained in the END register. The routine prepares for operation by clearing all locations in the range to 00_{16}. Next, the first location is set to FF_{16} and then a check is made of all address- related locations in the range to determine if any of them have been altered by the writing of the FF_{16}. After all locations in the range that are address-related to the first location have been checked, the program resets the first location to 00_{16}. Then the next sequential location in the range is set to FF_{16}, and the address- related locations checked. The test proceeds until all locations in the specified range have been set to FF_{16}, and the respective address-related locations checked.

For the purposes of this test, an address is said to be related to a second address if it differs from it by only one bit (in any bit position). The test checks all possible address-related combinations as long as a generated address does not fall outside the specified range.

If an addressing error is found the test stops, the EM-188 emits three beeps signifying an error, and the display shows the erroneous data and its address. At this point the operator has three options:

1. Depress EXAM to display the data the diagnostic routine expected to read. Release EXAM to return the erroneous data back to display.
2. Depress INC to continue testing. If additional problems are found, the test will stop and any of the options listed may be taken again.
3. To terminate the test depress RESET, or RUN or any of the mode selection keyswitches.

After all locations in the specified range have been tested, the EM-188 emits one short beep and repeats the test. To exit this code function at any time, depress RESET.

CODE A4 - ALL TESTS AND REPEAT

Code Function A4 executes the A1, A2, and A3 diagnostic functions in sequence, then emits a short beep and repeats. The test may be terminated by depressing RESET. In the event that an error is found, the operator may respond in any of the ways described for the individual diagnostic functions.

CODE A5 - ALL TESTS AND STOP

Code Function A5 executes the A1, A2 and A3 diagnostic functions in sequence, emits a short beep and stops. In the event an error is found, the operator may respond in any of the ways described for the individual diagnostic functions. When the test is complete, the displays will read CODE A5 and the Trace Memory will contain a record of the last 251 bus transfers.

CODE A6 - 00/FF DATA TEST

Code Function A6 is identical to the Code Function A1 except the function stops after a single pass through the test. When the test is complete, the displays will read CODE A6 and the Trace Memory will contain a record of the last 251 bus transfers.

CODE A7 - ROTATE 1's

Code Function A7 is identical to the Code Function A2 except that the function stops after a single pass through the test. When the test is complete, the displays will read CODE A7 and the Trace Memory will contain a record of the last 251 bus transfers.

CODE A8 - ADDRESS TEST

Code Function A8 is identical to the Code Function A3 except the function stops after a single pass through the test. When the test is complete, the displays will read CODE A8 and the Trace Memory will contain a record of the last 251 bus transfers.

7-2 GROUP B: OSCILLOSCOPE LOOPS

The Oscilloscope Loop Functions are a group of functions that provide several types of repetitive stimuli to a target system. They provide repetitive waveforms in the target system hardware that may easily be examined at various circuit points with an oscilloscope or other test equipment. Some of the functions are also useful as stimulus routines for Signature Analysis testing.

NOTE: ADDR register referred to below is "Hex Keyswitch - #6."

CODE B1 - REPETITIVE MEMORY READ

This function repetitively reads the single memory location addressed by the **ADDR register.** These address, data and RD signals are all shown on the Display. An EM- 188 configured for an 8085 processor reads the location about 30,000 times per second with a 5 MHz clock. Depress RESET to exit this function.

CODE B2 - REPETITIVE MEMORY WRITE

This function repetitively writes the data contained in the DATA register to the single memory location addressed by the **ADDR register.** These address, data and WR signals are all shown on the Display. An EM-188 configured for an 8085 processor writes the data to the addressed location about 30,000 times per second with a 5 MHz clock. Depress RESET to exit this function.

CODE B3 - REPETITIVE I/O READ

This function repetitively reads the single I/O port location addressed by the **ADDR register.** The address, data and RD signals are all shown on the display. An EM- 188 configured for an 8085 processor reads the location about 30,000 times per second with a 5 MHz clock. Depress RESET to exit this function.

CODE B4 - REPETITIVE I/O WRITE

This function repetitively writes the data contained in the DATA register to the I/O location addressed by the **ADDR register.** These address, data and WR signals are all shown on the Display. An EM-188 configured for an 8085 processor writes the data to the addressed location about 30,000 times per second with a 5 MHz clock. Depress RESET to exit this function.

CODE B5 - CONTINUOUS ADDRESS INCREMENT

This function places the EM-188 in a special mode in which it outputs successive addresses from 0000_{16} to $FFFF_{16}$ at a very high rate. Internal to the EM-188 this is accomplished by forcing a NOP instruction to the processor on every fetch cycle. Externally the EM-188 appears to be doing a fetch cycle at each address at the full speed of the processor (as determined by the clock frequency).

In this mode, the processor does **not** respond to target system WAIT commands.

The Continuous Address Increment function is used to check out address decoding networks in hardware systems and as a stimulus for signature analysis troubleshooting.

It is possible to obtain a sync pulse for triggering an oscilloscope or a signature analyzer from either the Breakpoint A or Breakpoint B output at the Auxiliary Connector; the output pulse occurs each time the processor reads from the breakpoint address. (The processor does not stop.)

Depress RESET to terminate the Continuous Address Increment mode.

CODE B6 - REPETITIVE MEMORY WRITE (DATA/DATA)

This function repetitively writes data to the address designated by the **ADDR register.** The data written is that contained in the DATA register except it is complemented every other time data is written. The address, data and WR signals all display. AN EM-188 configured for an 8085 processor writes data to the addressed location about 30,000 times per second with a 5 MHz clock.

CODE B7 - Repetitive I/O WRITE (DATA/DATA)

This function repetitively writes data to the I/O port designated by the **ADDR register.** The data written is that contained in the DATA register except it is complemented every other time data is written. The address, data, I/O and WR signals all display. An EM-188 configured for an 8085 processor writes data to the addressed location about 30,000 times per second with a 5 MHz clock. Depress RESET to exit this function.

CODE B8 - TOGGLE SOD LINES (8085 ONLY)

Executing this function causes the EM-188 to set the SOD line (PIN 4 of the 8085) alternately low and high. An Em-188 configured for an 8085 processor operating with a 5 MHz clock (10 MHz crystal) cycles the SOD line low-to-high about 110,000 times per second. (The 8080 processor does not have a SOD pin.) Depress RESET to exit this function.

7-3 GROUP C: MEMORY LOAD AND DUMP

CODE C1 - LOAD TARGET FROM FRONT PANEL PROM

The Code Function C1 transfers data from the Front Panel Diagnostic PROM to the target system. This routine requires the user to specify the destination address range in the target system by entering the first address of the range in register BEG and the last address of the range in register END. To use this Code Function, first enter the appropriate address values in the BEG and END registers, then start the routine. The routine will transfer bytes from the Front Panel PROM into the target address space with the first location in the PROM transferred to the first address of the specified range. After the transfer is complete, the Trace Memory contains a record of the last 251 cycles of the transfer.

CODE C2 - VERIFY TARGET AGAINST FRONT PANEL PROM

The Code Function C2 compares the Front Panel PROM with the address range specified by the user. The address range should be specified using the BEG and END registers in the same manner as described for C1.

CODE C3 - LOAD TARGET FROM SERIAL LINK (DOWNLOAD)

The Code Function C3 transfers hex data from the serial RS-232C input to the target system. The data to be entered must first be converted into the Intel MDS* format, which is an ASCII-hexadecimal format.

The destination address range is specified by the incoming data and need not be specified in the BEG and END registers. Furthermore, if the data is properly received, the BEG and END registers contain the low and high limits of the loaded data, regardless of the initial register settings. In addition, the limits will always be correct even if non-contiguous data is loaded.

To use this Code Function, connect the RS-232C input to the source of information, start this routine and then enable the source to "download" the appropriate data. During the transfer, note the displays showing the data being loaded. If there are no errors, the end-of-file record completes the transfer and the displays contain CODE C3. The Trace Memory contains a record of the last 251 cycles of the transfer.

Error Codes - During the data transfer process, various types of errors can occur. If an error occurs the Diagnostic Emulator emits three beeps and displays the address and erroneous data. Once an error is detected the transfer process is aborted and may not be resumed. The C3 Code Function Error Codes are listed in Table 7-3.1.

* *MDS is an Intel Trademark.*

Table 7-3.1. C3 Code Function Error Codes

CODE	DESCRIPTION
01	Framing Error. The serial data character is not properly framed by start and stop bits. This error may be caused by an incorrect setting of the baud rate selector or by noise on the transmission link.
02	Overrun Error. This error may occur if the processor is operated with an extremely low clock frequency while receiving data at high baud rates.
11	Non-Hexadecimal Character Received. This error indicates that a hex character was expected at some point, but a non-hex character was received.
12	Sum-Check Error. In the Intel MDS format each record contains an eight-bit check-sum to ensure data integrity. If this sum is incorrect, this error code is given.
16	Non-Zero Record Type. If the record type byte is other than zero (except for the end of file record), this error is signaled.
21	Target Memory Write Error. If an attempt is made to load data to an area of memory containing no RAM or faulty RAM, this error occurs. This error is detected by doing a read-back-check of each location as it is stored, and recording both the write cycle and the read cycle in Trace Memory. After the error occurs register ADDR will contain the address of the faulty location while register DATA will contain the data the EM-188 attempted to store.

CODE C4 - DUMP TARGET TO SERIAL LINK

The Code Function C4 transfers data from a selected area of Target Memory to the serial RS-232C output. The data being output is ASCII-Hexadecimal and is compatible with the Intel MDS format. The use of this function requires the user to specify the address range. The BEG register contains the starting address while the END register contains the address of the last location to be output.

To use this Code Function, first specify the address limits, next prepare the receiving device to accept data, then start the transfer by executing CODE C4. During the transfer the display shows the address and data currently being transmitted. When transmission is completed, the displays show CODE C4 and the Trace Memory contains a record of the last 251 cycles.

The rate of transfer can be controlled by the receiving device. If enabled by the Option Switchs (with position 3 open), the CTS line (Clear-to-Send) can prevent output if held in the marking (negative) condition. In the spacing (positive) condition, output speed is determined by the baud rate selected.

Each record is followed by a carriage return, line feed and two null characters.

CODE C5 - LOAD OVERLAY RAM FROM TARGET MEMORY

The Code Function C5 transfers data from a selected area of target memory space to the equivalent area in Overlay Memory. To use this Code Function the Overlay memory must first be located at the proper address by rotating the thumb-wheel switch. The Overlay is then enabled by setting the selector switch to the appropriate position. Then the BEG and END registers are set to the range of addresses over which data is to be transferred. The last step is to call the Code Function to execute the transfer. While executing, the displays show the address and data. When data transfer is completed, the displays show CODE C5 and the Trace Memory contains a record of the last 251 cycles.

If a non-verify occurs during the transfer, the Diagnostic Emulator emits three beeps and temporarily halts the transfer. The error may be skipped and the transfer resumed by depressing INC, or the operation may be aborted by depressing a mode select keyswitch, such as CODE. While the operation is halted, the address and the data that failed to verify is shown on the display. By depressing and holding the EXAM keyswitch, the correct target data may be displayed.

CODE C6 - VERIFY RAM OVERLAY WITH TARGET MEMORY

The Code Function C6 compares data from a selected area of Target Memory to the equivalent area in Overlay Memory.

For information on the operation of this function, see Code C5.

CODE C7 - VERIFY TARGET WITH SERIAL LINK

The Code Function C7 is nearly identical to the C3 Code Function. It differs in two respects:

1. Data is not stored to target memory but only verified.
2. A non-verify results in Error 22 and the transfer is aborted. Register ADDR will contain the address of the non-compare while register DATA will contain the data that was supposed to be in the target memory location.

CODE C8 - FILL MEMORY WITH DATA

The Code Function C8 is used to fill a block of target memory or RAM Overlay with the same data, usually all one's (FF) or all zeros. To use this Code Function, set the BEG and END registers to the range of target memory or RAM Overlay to be filled, load the DATA Register with the data to be stored then execute CODE C8. The Display shows the transfer as it takes place. After transfer is completed the display shows CODE C8 and the trace contains a record of the last 251 cycles of the transfer.

If a location fails to store the correct data, the Diagnostic Emulator emits three beeps and temporarily halts the fill operation. The error may be skipped and the transfer resumed by depressing INC, or aborted by depressing a mode select keyswitch such as TRACE. While the operation is halted the address and the data that failed to verify are shown on the Display. By depressing and holding EXAM the correct data (which was in DATA) may be displayed.

CODE C9 - VERIFY MEMORY WITH DATA

The Code Function C9 compares a block of target memory or RAM Overlay with the byte in register DATA. See the explanation of CODE C8 for the operation of the function.

7-4 GROUP D: MISCELLANEOUS

CODE D0 - CLEAR INTERRUPT ENABLE FLIP-FLOP

The interrupt enable flip-flop of the emulation processor is cleared so that the next time the processor starts running, interrupts will not be allowed. There are no parameters for this Code Function.

CODE D1 - SET INTERRUPT ENABLE FLIP - FLOP

The interrupt enable flip-flop of the emulation processor is set so that the next time the processor starts running, interrupts will be allowed. There are no parameters for this Code Function.

CODE D2 - DISPLAY CLOCK FREQUENCY

This Code Function is a routine that determines the clock frequency of the emulation processor by comparing the instruction execution rate of the processor with the EM-188 internal 1.2 KHz reference frequency. The internal reference frequency is derived from the crystal controlled UART clock. The frequency is displayed on the ADDRESS display and is given in kilohertz. For example, an 8085 operating with an 8.0 MHz crystal will have a 4.0 MHz clock rate; executing the CODE D2 function will display 4000 (kilohertz) on the ADDRESS displays. The result is accurate to about ±.01% (the accuracy of the UART crystal) with either the 8085 or 8080 Emulator Pods in use. There are no parameters for this Code Function.

CODE D3 - DISPLAY PROM/ROM SIGNATURE

The purpose of this Code Function is to provide a convenient way of verifying that all bits in a PROM or ROM are correct. The routine operates by reading each 8-bit byte in a specified range and shifting the bits into a firmware implemented feedback shift register. By this means, the routine calculates a 16-bit check value that is displayed as a 4-digit hexadecimal signature on the ADDRESS display. This signature has a very high probability (.9998) of being unique for any given bit pattern in a ROM.

A PROM or ROM signature is obtained by setting the first address of the ROM in the BEG register and the last address of the ROM in the END register; then execute the routine. The Code Function routine will calculate and display the ROM signature. If the correct signature has been obtained previously with a known good ROM, then the ROM under test is good if it has the same signature.

There is a technique that may be used to create PROMs or ROMs whose signatures are zero. For the method to work, the last two locations of the ROMs must be unused. Proceed as follows:

1. Program a PROM with the desired information, making sure that the last two bytes are zeros.
2. Determine the signature of the PROM using the CODE D3 function.
3. Program a new PROM with the desired information, but replace the last two bytes, which previously were zero, with the bit pattern of the signature obtained in Step 2.

Now, while the signature of the new PROM is being calculated, the routine will arrive at a point just prior to processing the last two bytes of the PROM and at that time, the shift register will contain the signature of the PROM as calculated in Step 2; entry of the last two bytes, containing the same bit pattern as that already in the shift register results in the shift register reaching a final value of zero when computation is complete. Thus, the PROM will have a signature of 0000.

CODE D4 - OUTPUT 50 NULS TO SERIAL PORT

This Code Function outputs 50 nul characters (00_{16}) to the serial port for the purpose of providing leader or trailer for users using punched paper tape as a data storage media. There are no parameters for this Code Function.

CODE D5 - CALL USER ROUTINE IN INTERNAL RAM AT 3000_{16}

CODE D6 - CALL USER ROUTINE IN INTERNAL RAM AT 3003_{16}

These two Code Functions provide a means for the user to transfer control to routines that have been entered into the EM-188 internal scratch pad RAM for various reasons. To make use of this feature, the user must understand the requirements of the programs that run in the EM-188 internal environment. See Section 8 - User Implemented Code Functions.

CODE D7 - CLEAR TRACE MEMORY

The EM-188 Trace Memory is cleared when power is applied as part of the power-on-reset operations. Code Function D7 is used to clear the Trace Memory at any other time. This routine does not use any parameters.

CODE D8 -

Not implemented.

CODE D9 - HALT CPU

Code Function D9 causes the CPU to execute a HALT instruction, thereby halting the CPU and stopping all bus activity. It is recommended that the CPU be halted any time that an EPROM is inserted into or removed from the Front Panel Diagnostic PROM Socket to avoid the possibility of crashing the internal control program of the EM-188. After the CPU has been halted, RESET must be used to resume normal operation. There are no parameters for this function.

7-5 GROUP E: CHANGE DEFAULT PARAMETERS

CODE E0 - DISABLE DISASSEMBLY (DEFAULT)

Code Function E0 is used to disable the disassembly software if it is in operation. See Section 6 - Disassembly.

CODE E1 - ENABLE DISASSEMBLY

Code Function E1 is used to enable the disassembly software. See Section 6- Disassembly.

CODE E2 - OUTPUT 72 CHARACTER LINES (DEFAULT)

Execution of this Code Function configures the Disassembly Software to output 72 character lines. See Section 6 - Disassembly.

CODE E3 - OUTPUT 80 CHARACTER LINES

Execution of this Code Function configures the Disassembly Software to output 80 character lines. See Section 6 - Disassembly.

7-6 GROUP F: INTERNAL OPERATIONS

CODE F - SET INTROSPECTION MODE

Execution of this Code Function sets the EM-188 so that its own internal address space becomes the "target system". After execution of the CODE F function, memory examine and store operations will be directed to the EM-188 internal address space; user implemented Code Functions may be executed in single-step mode (with some qualifications) and other internal operations performed. See Section 8 - User Implemented Code Functions.

SECTION 8

USER IMPLEMENTED CODE FUNCTIONS

8-1 OVERVIEW

The EM-188 Diagnostic Emulator has a low-insertion-force socket on the front panel that is designed to accept EPROMs similar to the Intel 2716 or 2732 devices. This front panel socket is called the Diagnostic PROM Socket. The purpose of the Diagnostic PROM Socket is to provide a means whereby the user may insert EPROMs programmed with his own diagnostic routines and execute them in a convenient manner from the EM-188 Keyboard. These user routines may perform almost any imaginable function. In most cases, the user will probably write special test or diagnostic routines to help test portions of the target system for which no Built-In Code Functions are provided. This discussion provides a view of the internal environment of the EM-188 from the programmer's perspective and is intended to provide the information needed by the user to write and debug his own Code Functions.

The programmer is already familiar with the environment of his own target system. He knows there is a 64K byte address space called the Memory Address Space and within this address space are various blocks of ROM, RAM and, in some systems, I/O control or data registers. In addition, there is a 256-byte I/O address space which, in most systems, contains the addresses of I/O devices.

The EM-188 also has an internal address space with its own ROM, RAM, and I/O. The EM-188 control program and Built-In Code Functions reside in this address space. Any EPROM plugged into the Diagnostic PROM Socket also appears in this internal address space. It is thus possible for User Code Functions to access all Diagnostic Emulator facilities and to function exactly as if they were factory- programmed.

However, the Code Function programs executing within the internal address space do not have direct access to the user's system target address space. If it is necessary for a Code Function to read or write to the external target system, it must do so in cooperation with the Diagnostic Emulator hardware circuits. Consequently, a rigidly defined routine must be executed to perform read or write operations to the target system. The Built-In Code Functions, as well as the EXAMINE and STORE routines, use EM-188 control program subroutines, and the user may also use these same subroutines to read and write to the target program address space.

8-2 INTERNAL ENVIRONMENT

The internal environment of the EM-188 contains ROM, RAM and I/O. The I/O devices of the EM-188 are memory mapped. Figure 8-2.1 shows an overview of the EM-188 internal address space.

Figure 8-2.1. EM-188 Internal Memory Map

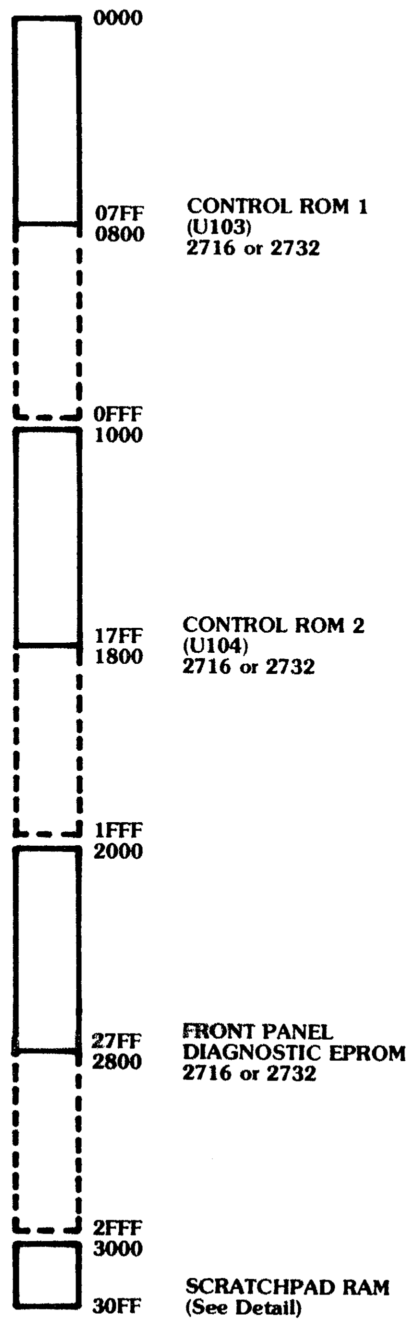


Figure 8-2.2. Map of Internal Scratchpad RAM

ADDRESS	Label	Size	Description	Bytes
3000	SCRAM		For use by user written code function programs. (96 Bytes)	
305F				
3060			Firmware scratchpad locations used by emulator control program. (97 Bytes)	
30C0				
30C1			Emulator firmware stack area. (42 Bytes)	
30EA				
30EB	RDE	16	Target Registers DE	(2 Bytes)
30ED	RBC	16	Target Registers BC	(2 Bytes)
30EF	RAF	16	Target Registers AF	(2 Bytes)
30F1	RHL	16	Target Registers HL	(2 Bytes)
30F3	RSP	16	Target Register SP	(2 Bytes)
30F5	RPC	16	Target Register PC	(2 Bytes)
30F7	RRIM	8	Target Register RIM	(1 Byte)
30F8	RSIM	8	Target Register SIM	(1 Byte)
30F9	RBEG	16	Begin Pointer	(2 Bytes)
30FB	REND	16	End Pointer	(2 Bytes)
30FD	RADR	16	Current Address of Test	(2 Bytes)
30FF	RDTA	8	Current Data	(1 Byte)
			TOTAL:	256 Bytes

3100 ↓ 33FF — Additonal Scratch RAM (See Sec. 8-2.3.)

8-2.1 ROM

The EM-188 has two sockets, located on the Keyboard circuit card, that accept EPROMs or ROMs that contain the control program for the unit. The circuit board connections are normally set up for EPROMs or ROMs having the Intel 2716 pinout; a jumper modification of the board allows use of the 4K byte 2732 as well. See Figure 8-2.1.

8-2.2 FRONT PANEL EPROM SOCKET

The EM-188 Front Panel EPROM Socket also accepts EPROMs of the 2716 or 2732 variety. A small switch located in the center of the socket selects the appropriate connections for either the 2K byte or 4K byte EPROM types. In the internal address space, the EPROM plugged into the Front Panel Socket will appear in the address range of 2000_{16} to $27FF_{16}$ (2716) or 2000_{16} to $2FFF_{16}$ (2732) in the internal address space. It is not possible to have this EPROM appear in the external (target) address space. See Figure 8-2.1.

8-2.3 SCRATCHPAD RAM

The EM-188 also contains a small amount of Scratchpad RAM in the internal address space that is used by the control program in keeping track of the status of the emulator and the emulation processor. The Scratchpad RAM resides in the internal address space at addresses 3000_{16} to $30FF_{16}$. Figure 8-2.2 shows a detail of the Scratchpad RAM. The first 96 bytes of the RAM are available to user- written Code Function programs.

The Scratchpad RAM also contains the area where the processor registers are saved each time the emulation processor pauses; also, the saved register values are restored each time the emulation processor begins to run. User implemented programs may obtain these register values or even alter them if desired. The user should carefully avoid altering any of the data contained in the firmware stack area or firmware scratchpad locations to avoid crashing the control program.

NOTE: EM-188's with serial #1181140 or higher have an additional 768 Bytes of Scratchpad RAM residing from 3100_{16} to $33FF_{16}$. This additional RAM area is available for general purpose use.

8-2.4 I/O DEVICES

The EM-188 control program, running in the internal address space, has access to various I/O registers associated with different components of the emulator. Some users may wish to create special programs that control the EM-188 components in a way different than that provided for by the standard software, and for that reason, this detailed information is provided. In most cases, however, the user will be able to obtain the desired result by using I/O handler subroutines that are already present in the EM-188 firmware. Section 8-6 provides information on the characteristics and use of the User Accessible Subroutines.

KEYBOARD: The state of the Keyboard keyswitches may be read by the processor at a series of eight addresses from 3400_{16} through 3407_{16}. Four keyswitches may be read at each of the input addresses as shown in Figure 8-2.3. A key depression causes the corresponding bit to go low as seen in the input data. For example, if key 9 is depressed, bit 1 of location 3402_{16} will be low. Bits 4, 5, 6 and 7 of all eight of the input ports see the same data; bit 4 in all locations will be low if a jumper on the Keyboard called the Option A jumber is installed. Bit 5 in all locations will be low if an 8085 Emulator Pod is attached to the Operator's Station; bit 5 will be high if an 8080 Emulator Pod is attached. Bit 6 will be low if the most recent system reset was caused by the power-on-reset circuitry; bit 6 will be high if the most recent system reset was caused by the RESET key or a reset command from the target system. Bit 7 will be high if any of the following keys are depressed: RUN, RUN-BKPT or STEP.

The user who decides to write software to directly read the Keyboard must be aware that there is no key debouncing or other processing of the key closure done by the hardware. Consequently, it is necessary to provide the keystroke debouncing, repeating key features or other special processing in the software that scans the Keyboard. There is a keyboard scan routine already in the EM-188 which may be accessed by the user that provides the most commonly needed features. See Section 8-6.

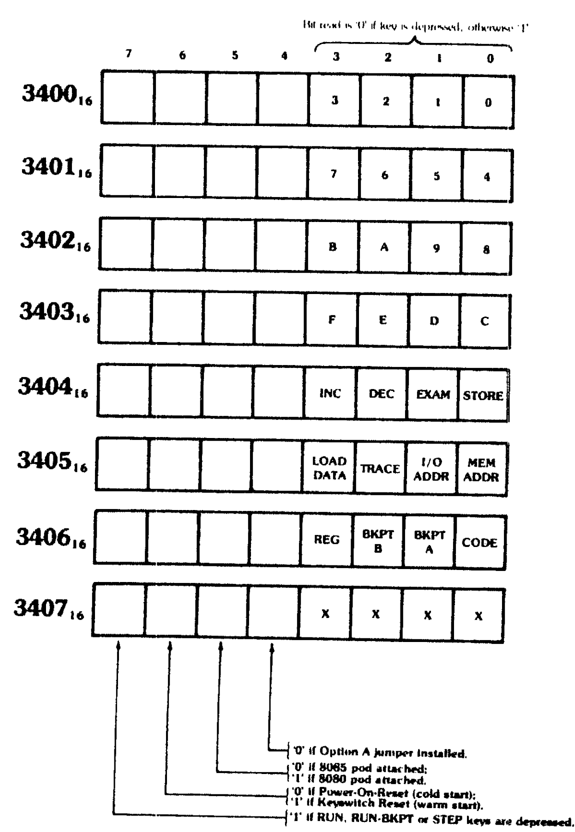


Figure 8-2.3. Keyboard Input Locations

SERIAL INPUT/OUTPUT PORT: The EM-188 Diagnostic Emulator contains circuitry that implements a full-duplex (two-way) serial Input/Output port that conforms to RS-232C requirements. The baud-rate, parity and character length of the data transmitted and received is set up by hardware switches. (See Sections 9-2 and 9-3). The nature and format of data transmitted is under the control of software. The software is able to send data to the serial output circuits, read data from the serial input circuits and test the status of the serial port circuitry via three ports as shown in Figure 8-2.4. Data is transferred to and from the serial port by means of a Universal Asynchronous Receiver-Transmitter (UART).

Data to be output through the serial port is written to the UART Data Write address. The data enters the UART transmit buffer register, and then enters the transmit shift register where it is shifted out in serial form bit by bit. New data may be written to the transmit buffer register as soon as the previous data has entered the transmit shift register and before it has completed the process of shifting out.

The UART Status Register (Figure 8-2.4) contains two bits which inform the software of the status of the transmitter registers as follows:

Bit 1, Transmit Buffer Empty, will be read as a '1' when the transmit buffer register may be loaded with another character. A '0' means that the transmit data register contains data that has not yet been moved into the transmit shift register.

Bit 2, End of Character, will go to '1' at the time that a character has shifted out of the transmit shift register. If there is another character waiting in the transmit buffer register, then bit 2 will immediately go to '0' as the new character enters the shift register to be transmitted.

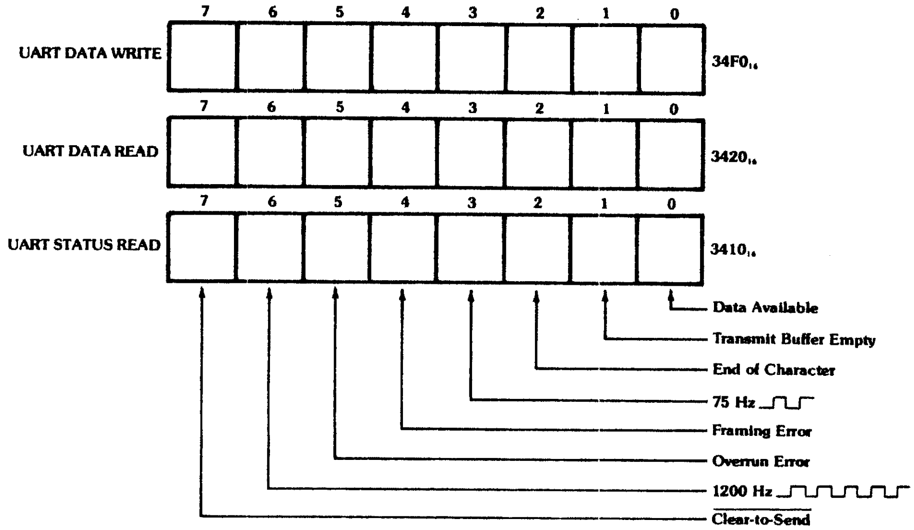
Data received by the EM-188 through the serial port is entered into the UART receiver shift register. When an entire character has been received, it is transferred to the receiver holding register and is then available to the software by reading the data at the UART Data Read address. Several status bits in the UART Status Register (Figure 8-2.4) give information about the received data as follows:

Bit 0, Data Available, goes to '1' when an entire character has been received and transferred to the receiver holding register. When the software reads the UART Data Read location, this bit is cleared to '0'.

Bit 4, Framing Error, goes to '1' if the received character has no stop bit at the expected location. This usually means that the transmitting device is sending characters of different length than the EM-188 is set up to receive. Noise may also cause this error.

Bit 5, Overrun Error, goes to '1' if a previously received character in the receiver holding register is not read by the CPU before another character is received and transferred into the holding register.

Figure 8-2.4. Serial Port Data and Status Locations



The clear-to-send input (Auxiliary Connector, Pin 5) is visible to the software as Bit 7 of the UART status register. This bit is '0' if clear-to-send is true (high); if clear-to-send is low or disconnected, this bit is '1'. (Note, however, that this bit may be forced to the '0' state by setting Option Switch 3 closed. See Section 9-2.)

Two other bits share the UART Status Register, but are not directly involved in the communications functions.

Bit 3, 75 Hz, is a 75 Hz square wave that is derived from the bit-rate-generator crystal oscillator. This bit is seen by the software as alternate '1' and '0' with a 13.33 mSEC full cycle.

Bit 6, 1200 Hz, is a 1200 Hz square wave that is derived from the bit-rate-generator crystal oscillator.

One additional output port is associated with the communications interface. This port controls the Request-to-Send (RTS) signal that is output on Pin 4 of the Auxiliary Connector.

	7	6	5	4	3	2	1	0	
RTS		X	X	X	X	X	X	X	$34E7_{16}$

The RTS output port is located at address $34E7_{16}$. Writing a '1' to Bit 7 of the port will set the RTS signal to its negative (marking, or OFF) state; writing a '0' sets the RTS signal positive (spacing or ON). Bits 0 through 6 of the port are "don't care" and have no effect.

HEXADECIMAL DISPLAYS AND TRACE MEMORY: The EM-188 Trace Memory is a 251-word by 32-bit memory whose primary function is to record each bus cycle that occurs to the target system. At any given time, a single word of the Trace Memory is selected by an 8-bit register called the 'XADDR' (trace index address) register. If, for example, the XADDR register contains a 43, then the next bus cycle that occurs will be written into location 43 of the Trace Memory. Immediately after the data is written, the XADDR register is incremented (by hardware) so that the current Trace Memory address becomes 44. If the emulation processor were executing a target program, each bus cycle is written into the Trace Memory and XADDR is incremented for the next cycle. When XADDR reaches its maximum value of FF_{16} and is again incremented, it overflows to 00_{16} so that the first location of the memory effectively follows the last location. Thus the Trace Memory may be viewed as a ring memory in which each additional bus cycle may be entered in the next position around the ring. Once the Trace Memory is full, each additional bus cycle simply overwrites the oldest bus cycle in the memory.

The Address and Data hexadecimal displays and the eight discrete Machine Cycle indicators are wired directly to the Trace Memory circuitry so that the current Trace Memory word (the word designated by the XADDR register) is always displayed unless the displays are explicitly blanked.

When the EM-188 is in the PAUSE mode, the internal control program has access to the Trace Memory and the displays by means of a set of five ports. See Figure 8- 2.5. The five ports are as follows:

XADDR This port gives access to the Trace Index Address register. The control program may read this location to obtain the current value of the XADDR register, and may store new values in the register. Storing a new value in XADDR will change the current trace word that is accessed and displayed on the EM-188 display panel.

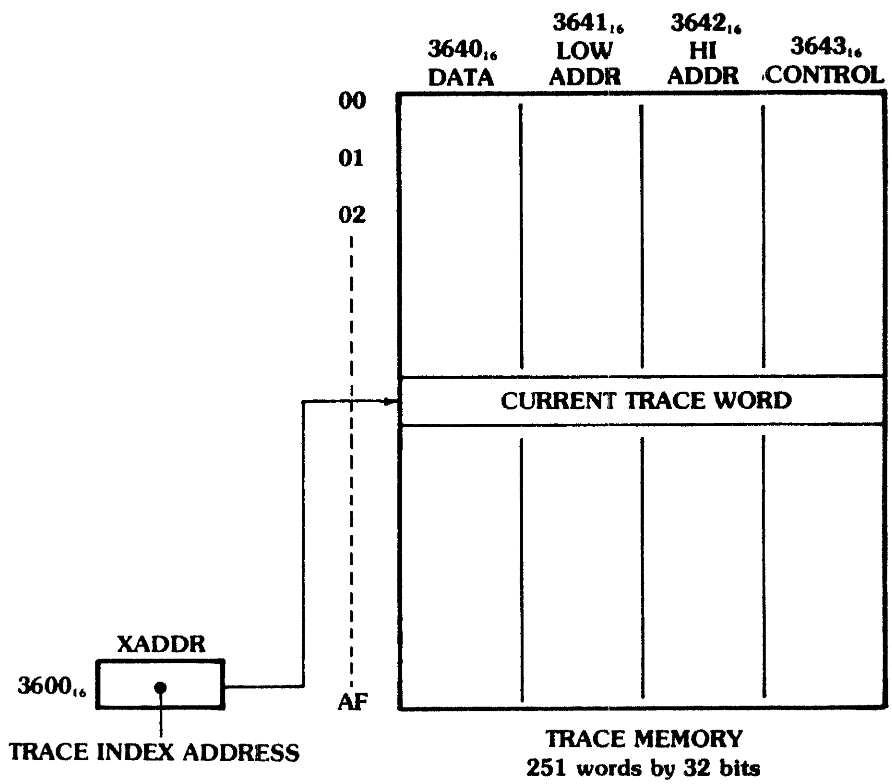
TDATA This port gives access to the eight-bit wide portion of the current Trace Memory word that records the data bus signals of each machine cycle. The control program may read this location to obtain the data portion of the current trace word, or may store new data to the data portion of the trace word.

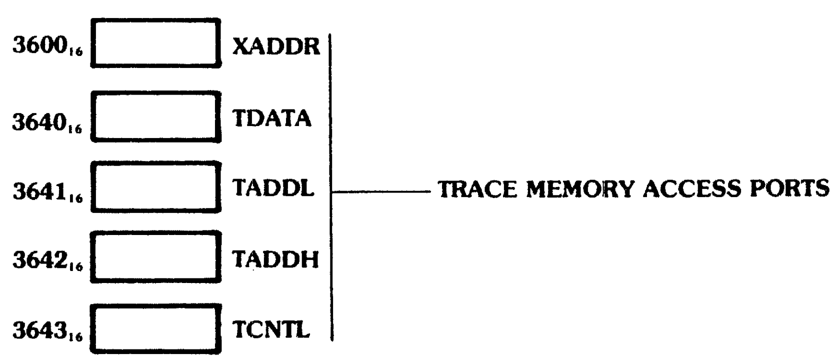
TADDL This port gives access to the eight-bit portion of the current trace word that records the low order eight bits of the address bus of each machine cycle. The control program may read or write this location.

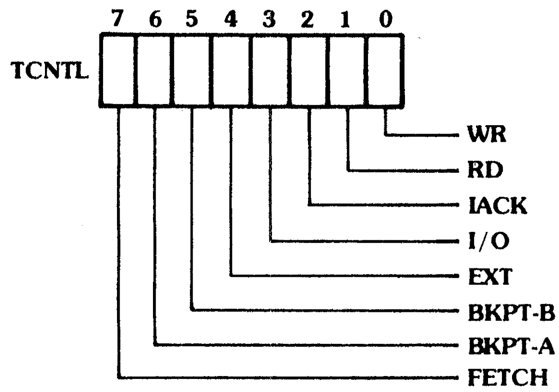
TADDH This port gives access to the eight-bit portion of the current trace word that records the high order eight-bits of the address bus. The control program may read or write this location.

TCNTL This port gives access to the eight-bit portion of the current trace word that records the control bits of each machine cycle. The control program may read or write this location. The control bits are arranged in the port as shown.

Figure 8-2.5. Trace Memory Access Ports

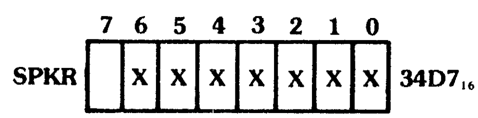






Any time that the control program writes new data into the Trace Memory, the data stored will immediately be seen on the appropriate displays.

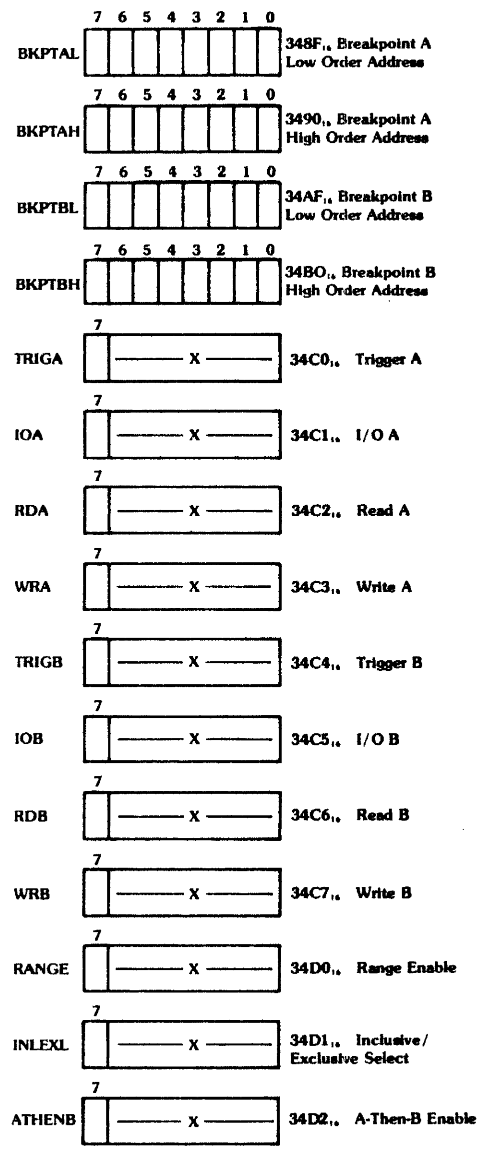
SPEAKER: The EM-188 incorporates a very small dynamic speaker that is located on the Keyboard printed circuit board. A port is provided to control the current to the speaker to generate tones or other sounds under software control. It is necessary for the software to generate the actual waveform to be output by the speaker; there is no tone generation hardware in the EM-188. The speaker output port is diagrammed below:



Writing a '0' to Bit 7 switches DC current to the speaker ON; writing '1' to Bit 7 switches the current OFF. Bits 0 through 6 of the port are "don't care" bits and have no effect.

HARDWARE CONTROLS: There are a few additional output ports in the EM-188 internal address space that are used for such functions as initiating the Binary Address mode, entering the "introspection" mode and other hardware features. Details of these ports are not contained in this manual.

BREAKPOINT COMPARATORS: A series of output ports is used to set up the Breakpoint Comparator address values and to control the desired operating mode. These ports are detailed below:



The ports BKPTAL and BKPTAH are used to set up the A Breakpoint address. The software should store the appropriate low and high order address bits to these ports. In the case of an I/O address which only has eight bits, the low order breakpoint address should be loaded; the high order breakpoint address port will have no effect.

The ports BKPTBL and BKPTBH are used to set up the B Breakpoint address in a manner analogous to that described above.

Two ports, TRIGA and TRIGB, are used to output pulses to the BKPT A and BKPT B output pins of the auxiliary connector (pins 12 and 13; see Figure 9-1.1). Storing a '1' to the bit 7 position of the ports results in a high level at the corresponding output pin. To generate an output pulse the software must store a '1' followed by a '0' to the bit 7 position of the port. Bit positions 0-6 of the ports have no effect on the system. Inclusion of these ports in the EM-188 system enables user programmed Code Functions to output trigger signals to external equipment such as oscilloscopes or signature analyzers.

Ports IOA and IOB are used to configure the breakpoint comparators to respond to either memory cycles or I/O cycles. Storing a '1' to the bit 7 position of either port will cause the associated breakpoint comparator to respond to I/O cycles with matching 8-bit low order address values. Storing a '0' to the bit 7 position of either port will cause the associated breakpoint comparator to respond to memory cycles with matching 16-bit address values. Bits 0 through 6 of these ports have no effect.

Ports RDA and RDB are used to enable the breakpoint comparators to respond to read cycles (either memory read cycles or I/O read cycles). The comparators will respond to read cycles if a '1' is stored to bit position 7 of the associated port.

Ports WRA and WRB are used to enable the breakpoint comparators to respond to write cycles (either memory write cycles or I/O write cycles) by storing a '1' to bit position 7 of the relevant port.

Notice that if '1' bits are stored to both the RDA and WRA ports, the A breakpoint comparator will respond to both read and write cycles. If a '0' is stored to both ports, the comparator will not respond to any cycles and is thus disabled. The B breakpoint comparator may be controlled in a similar manner.

Three ports, RANGE, INLEXL and ATHENB are used to configure the breakpoint circuitry for several special operating modes (see Section 4-1.4). Any address in the range from A to B may be detected by writing a '1' to bit 7 of the INLEXL port. In order to include both end points of the range (address = A and address = B), it is recommended that both the A and B comparators also be enabled for read and write cycles. Due to limitations of the hardware design, it is not possible to break on a range only on the occurrence of a read cycle or only on the occurrence of a write cycle.

With a '1' stored to the INLEXL port, the inclusive range (everything from A to B) is detected. With a '0' stored to the INLEXL port, the exclusive range (every address outside of the range A to B) is detected. Again, it is required that the A and B comparators also be set up to detect read and write cycles to ensure that both end points are included in the range. Also, make sure that the RANGE circuitry is enabled ('1' to RANGE part).

The ATHENB port is used to set up sequential operation of the comparators so that a breakpoint stop signal is generated only when the B address is encountered after the A address has previously been encountered. To set up this mode, store a '1' to the bit 7 position of the ATHENB port. Also, make sure that the RANGE circuitry is disabled ('0' to RANGE port) and the A comparator is disabled ('0' to RDA and WRA ports) but the B comparator is enabled ('1' to RDB and WRB ports).

8-3 ENTRY TO USER CODE FUNCTIONS

The Code Functions that are built in to the EM-188 are all called with Keystroke sequences that begin with one of the letter keys, such as CODE A1, CODE C4 and CODE D2. The Code Functions that use the decimal digit keys (0-9) are reserved for calling user programmed Code Functions. The Keystroke sequences that are used to transfer control to user Code Functions are as follows:

Key Sequence	**Transfer Address**
CODE 0	2000_{16}
CODE 1	2003_{16}
CODE 2	2006_{16}
CODE 3	2009_{16}
CODE 4	$200C_{16}$
CODE 5	$200F_{16}$
CODE 6	2012_{16}
CODE 7	2015_{16}
CODE 8	2018_{16}
CODE 9	$201B_{16}$

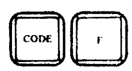
Thus, each of the key sequences has associated with it an entry address in the address space assigned to the Diagnostic Prom socket. It is the responsibility of the user to properly code the EPROM so that the desired actions occur for each entry address. See the examples given in Section 8-8.

8-4 INTRODUCTION MODE

The EM-188 Diagnostic Emulator has been designed with a special feature that is primarily intended as an aid to testing and debugging Code Function programs that have been programmed into EPROMs and plugged into the Diagnostic Prom socket. This special feature is the "Introspection Mode" in which the EM-188 is caused to turn its attention to its own internal address space. In this way, the user may examine and store to the internal address space and, with certain limitations, may single step programs that execute in the internal address space.

8-4.1 CODE F

The introspection mode is entered by the key sequence:



After entering the CODE F mode, the user may examine or alter the internal memory space, step or run programs in the internal memory space, and review the contents of the trace memory after program execution. Breakpoints may also be used to halt program execution at appropriate internal addresses. The RESET key returns the EM-188 to normal operation.

8-5 GETTING TO AND FROM THE TARGET SYSTEM

The EM-188 Control Program, together with the built-in diagnostic routines and any user-programmed code functions, executes within the EM-188 internal "protected" address space. As a consequence, programs in this internal address space do not have direct access to the target address space, but must make use of special hardware in the EM-188 logic to make the target address space accessible. Code Function Programs may have a requirement from time to time to do one of the following things:

1. Read from or write to a location in the target address space.
2. Read from or write to an I/O port address in the target address space.
3. Go to and begin executing a program residing in the target address space.
4. Return from running a program in the target address space to a program (user Code Function routine) in the internal address space.

The following sections give detailed information on these functions.

8-5.1 EXAMINE AND STORE

The simplest method of reading and writing data to the target system is to use subroutines that exist in the EM-188 control program. Four subroutines are provided, as follows:

STM Store the data contained in the accumulator to the target address specified in register-pair HL.

EXM Load the accumulator from the target address specified in register-pair HL.

STIO Store the data contained in the accumulator to the target I/O port specified by register C.

EXIO Load the accumulator from the target I/O port specified by register C.

The entry addresses of these subroutines (and other useful subroutines) are given in Section 8-6, User Accessible Subroutines.

The four subroutines shown above operate by performing a read or a write operation to the specified address after commanding the EM-188 hardware to make the target address space accessible during the transfer interval. The machine code listing of the EXM subroutine is shown below.

```
;
; Subroutine to read data from the target address
; specified by HL. Data is returned in A. The
; target read cycle is recorded in the trace memory.

EXM        PUSH B
           MVI A, 0E0H        ; Delay value
           STA DELAY          ; Command to hardware
           MOV B, M           ; Read from target
;
; The data read during the target memory read
; cycle above is automatically recorded in the
; trace memory. The trace memory bookkeeping
; values must be updated.
           LDA XADDR          ; Get trace index
           STA XBASE          ; Save
           DCR A
           STA TRACE
;
; Put data in place and return to caller.
           MOV A, B
           POP B
           RET
```

8-5.2 PAUSE to RUN

The EM-188 Control Program (firmware) is normally in control of the emulator when in the PAUSE mode. Depressing the RUN or RUN BKPT Key causes the control program to execute a sequence of operations that will load the processor registers with the values that had previously been saved (when the EM-188 last entered the PAUSE mode) and then does a coordinated jump to the target system program. The EM-188 hardware will switch to the target address space at the correct time for execution of the first instruction. It is possible for a user-written Code Function program to jump to a target system program in the same manner. This facility allows a user Code Function to place programs into the target address space (either user RAM or Overlay RAM) and then transfer control to that program. The target system program will then proceed to execute from within the target address space in a normal manner.

The best way for the user to transfer control to a target system program is to use the existing EM-188 internal routine. The following steps are suggested:

1. Set up the user register save locations (addresses $3OEB_{16}$ to $3OFF_{16}$ in the scratchpad RAM) as desired. In particular, be sure to set the target program counter (RPC) location to the correct starting address. Other registers may be used, if desired, as a means of passing parameters to the target program.
2. If the target program needs parameters or data from the internal program, then transfer the data to the target RAM using the STORE subroutine as appropriate.
3. Perform the transfer of control to the target program by jumping to the RUN routine at address $IO4F_{16}$.

The RUN routine will load the processor registers, do the required coordination of the EM-188 hardware, and start the target program running with the desired register values initialized.

8-5.3 RUN to PAUSE

When a program is executing in the target address space and it is desired to transfer control into the internal software, there are only three ways available to cause this to occur. They are:

1. Reset the system.
2. Press the STEP key.
3. Cause a breakpoint to occur, either with one of the breakpoint comparators or by means of the external breakpoint input connection.

The first two methods are commonly used during manual operation; the third method may be used during manual operation or a sort of automatic operation in which it is desired that a Code Function set up the conditions to enable a target program to get back to the internal environment when it so desires.

The following steps are suggested as a method that will allow a program executing in the target system address space to re-enter the internal address space:

1. A Code Function program sets up one of the breakpoint comparators to monitor a pre-arranged address in the target memory space.
2. The Code Function program sets up the re-entry jump address so that when the pre-arranged address is encountered and the breakpoint occurs, control will be given back to the Code Function program instead of the Keyboard Scan routine.
3. The Code Function program gives control to the target system program which then begins running.
4. At this point, the operator must depress the RUN BKPT Key to arm the breakpoint system.
5. When the target system program is ready to return control to the Code Function program, it accesses the pre-arranged address and causes the breakpoint to occur. After the EM-188 software has saved the processor registers, it will jump to the address specified in Step 2 (above) and the Code Function program may proceed.

8-5.4 RE-ENTRY JUMP

Some applications require that the EM-188 control software transfer control to a user program each time emulation of the target program is halted. Such a program might be a "soft shutdown" program that prevents damage to the target system when execution is halted. (See Section 9-8, Soft Shutdown.) The EM-188 has the flexibility required to give control to a user-written subroutine each time the RUN to PAUSE sequence of the emulator is executed. Normally, this subroutine would be programmed into an EPROM and inserted into the front panel socket of the EM- 188.

In normal operation, the EM-188 executes an internal RUN to PAUSE routine each time the target program is halted. This routine first saves the processor registers in the scratchpad RAM save area, sets up the display to show the correct data, and finally goes to the Keyboard input routine to determine the next action required. Before going to the Keyboard routine, however, the RUN to PAUSE routine examines location $30A0_{16}$ to see if it contains a jump instruction op-code ($C3_{16}$). If it does, the EM-188 will regard the jump instruction as the first instruction of a user-supplied subroutine, and will call the subroutine. (The EM-188 calls the address of the jump instruction which then jumps to the main body of the subroutine.)

The user-supplied subroutine will usually be located in the front panel EPROM, but may also be located in the user portion of the internal scratchpad RAM as the following example illustrates.

The following small program may be entered from the Keyboard of the EM-188. It causes the EM-188 to beep each time it transfers from RUN to PAUSE. Enter the program with the following steps:

1. Reset the EM-188, then execute CODE F to place the emulator in "introspection" mode.
2. Enter the jump instruction:
   at $30A0_{16}$ enter $C3_{16}$;
   at $30A1_{16}$ enter 00_{16};
   at $30A2_{16}$ enter 30_{16}.

   These three bytes constitute a jump instruction to location 3000_{16} in the internal address space. Location 3000_{16} is the first location of the user portion of the scratchpad RAM.
3. At memory address 3000_{16}, enter the following two-byte program:
   at 3000_{16} enter DF_{16};
   at 3001_{16} enter $C9_{16}$.
4. Reset the emulator to exit the "introspection" mode and proceed to operate the emulator. Note that each time the emulator transfers from RUN to PAUSE, the beeper will sound.

In most practical cases, the user subroutine will be located in the front panel EPROM instead of RAM as was done in this example. Also, the jump instruction may be easily written into addresses 30A0 through 30A2 by a Code Function program also residing in the EPROM. Executing the Code Function will enable the user subroutine and a second Code Function could be written to disable the subroutine by changing the jump instruction op-code to 00_{16} (or to any other code except $C3_{16}$).

8-6 USER ACCESSIBLE SUBROUTINES

The EM-188 Control Program contains handlers and subroutines which may be used by the user in constructing his own Code Functions. The entry addresses and functions of the routines are summarized in the following table.

Table 8-6.1. User Accessible Subroutines

ADDRESS	NAME	DESCRIPTION
1000_{16}	STM	Store the data contained in the accumulator to the target address specified in register-pair HL. (Flags are altered.)
1003_{16}	EXM	Load the accumulator from the target address specified in register-pair HL. (Flags and accumulator may be altered.)
1006_{16}	STIO	Store the data contained in the accumulator to the target I/O port specified by register C. (Flags may be altered.)
1009_{16}	EXIO	Load the accumulator from the target I/O port specified by register C. (Flags and accumulator may be altered.)
$100C_{16}$	STATE0	This routine places the Diagnostic Emulator in the TRACE VIEW mode and makes it ready to accept input from the Keyboard. This routine does not return to the calling program. (See XECUTE routine.)
1013_{16}	TAB	This subroutine outputs ASCII space characters (20_{16}) until the total number of printable characters output since the last carriage return character ($0D_{16}$) is equal to the number specified in the byte following the call to this routine. This routine is used to format output displays to CRT or printer terminals. No registers are altered.
1016_{16}	CRLF	Outputs a line ending sequence to the RS-232C port which consists of a carriage return ($0D_{16}$) line feed ($0A_{16}$), and two null characters (00_{16}).
1019_{16}	BYTE1	Convert the contents of the accumulator to two ASCII characters representing the value in hexadecimal, then outputs these characters to the RS-232 port. No registers are altered.
$101C_{16}$	LWLMIT	Compares HL to the BEGIN address; the subroutine returns with CY = 1 if HL $\geqq$ BEGIN. (Registers A and F may be altered.)
$101F_{16}$	HILMIT	Compares HL to the END address; the subroutine returns with CY = 1 if HL $\geqq$ END. (Registers A and F may be altered.)
$12C6_{16}$	CODEFN	Executes the built-in Code Function designated by the contents of the accumulator. E.G., if the accumulator is loaded with $A8_{16}$, the EM-188 will execute Code Function A8. If the Code Function completes successfully, this subroutine will return to the calling program.

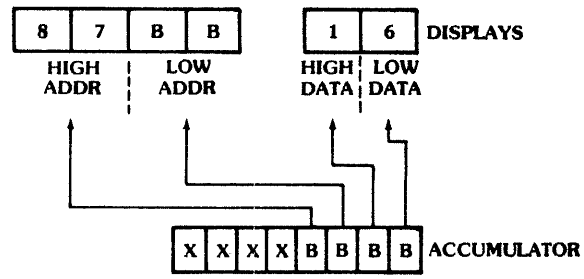
ADDRESS	NAME	DESCRIPTION
1022_{16}	DLYCYC	Delays until a low-to-high transition of the 1200 Hz internal clock occurs. (No registers are altered.)
1025_{16}	XECUTE	This routine returns program control to the standard Diagnostic Emulator firmware and readies it for Keyboard input. The display, however, remains as it is until the operator depresses another keyswitch. (See STATE0 routine.) This routine does not return to the calling program.
1028_{16}	UKBDSN	User's Keyboard scan routine. Data representing the keyswitch depressed is returned in the accumulator. In addition, if the keyswitch depressed is one of the hexadecimal numeric keys, the carry bit (CY) is set to one (true) when the subroutine returns. This routine ignores the RUN, RUN BKPT and STEP keyswitches. The table below shows the data returned in the accumulator for each keyswitch depression.

KEYSWITCH	DATA
RESET	*
RUN RUN BKPT STEP	**
CODE	18_{16}
BKPT A	19_{16}
BKPT B	$1A_{16}$
REG	$1B_{16}$
MEM ADDR	14_{16}
I/O ADDR	15_{16}
TRACE	16_{16}
LOAD DATA	17_{16}
STORE	10_{16}
EXAM	11_{16}
DEC	12_{16}
INC	13_{16}

KEYSWITCH	DATA
0	00_{16}
1	01_{16}
2	02_{16}
3	03_{16}
4	04_{16}
5	05_{16}
6	06_{16}
7	07_{16}
8	08_{16}
9	09_{16}
A	$0A_{16}$
B	$0B_{16}$
C	$0C_{16}$
D	$0D_{16}$
E	$0E_{16}$
F	$0F_{16}$

* When RESET is depressed, the User's Code Function is aborted and the EM-188 reinitialized.

** Keyswitches RUN, RUN BKPT and STEP are ignored by this routine.

ADDRESS	NAME	DESCRIPTION
$102B_{16}$	DSPCTL	Display Control. This subroutine uses the low-order four-bits of the accumulator to control the blanking of the display digits. This is shown in the example below.  If a bit of the accumulator is a one when this subroutine is called, then the display digit or pair of digits corresponding to that bit illuminate(s). The accumulator and flags may be altered by this routine.
$102E_{16}$	SI	Serial Input. Serial data entered at the serial port is returned in the accumulator. Bit seven is always zero. Registers A and F may be altered. When this routine is called, the RTS line (request-to-send) automatically goes high. The RTS line goes low again whenever the XECUTE routine is entered. The accumulator and flags may be altered by this routine.
1031_{16}	RHB	Read-Hex-Byte. This subroutine obtains two ASCII characters from the serial port. It checks to see if the characters represent valid hexadecimal digits. If they are, they are then converted to an eight-bit binary number and the subroutine returns to the calling program with this result in the accumulator. If any characters are received that are not valid hexadecimal characters, an error code (EC 11) is displayed and the Diagnostic Emulator must be reset to proceed. This subroutine alters the accumulator, flags and D register. The D register is used to accumulate a sum check of the data received. (All the eight-bit binary values are added into register D, with overflows ignored.)
0018_{16}	BEEP	A subroutine to beep the speaker located on the Keyboard printed circuit card. No registers are altered. This subroutine may also be called with an RST 3 instruction.
0020_{16}	DECXAD	A subroutine to decrement the XADDR register, which is the Trace Memory position register. When the XADDR register is decremented, the Trace Memory word preceding the current word is displayed. (That is, the previous Machine Cycle display, including address, data and control bits are seen on the Display Panel.) The flags may be altered by this subroutine. This subroutine may also be called with an RST 4 instruction.

ADDRESS	NAME	DESCRIPTION
0028_{16}	SOUT	Data in the accumulator is sent to the serial port. In addition, this subroutine affects a one byte counting location that is reset to zero each time a carriage return code is output ($0D_{16}$) and is incremented by one each time a printable ASCII character is output. This counting location is also affected and examined by the TAB routine. This subroutine may also be called with an RST 5 instruction.
0030_{16}	IMSO	The data byte following the subroutine call or restart code is sent to the serial port. No registers are altered. This subroutine may also be called with an RST 6 instruction.

8-7 INTERRUPTS

Target system interrupts are invisible to programs executing in the EM-188 internal environment. For this reason it is not possible to write Code Function programs that directly work with or test the user's interrupt system. Nevertheless, it is possible for a Code Function program to test or work with interrupts as follows:

1. The internal Code Function program, when it begins executing, first copies the interrupt portion of the routine to target system RAM. (If no RAM is available in the target system, the RAM Overlay may be used.)
2. The Code Function program sets up one of the breakpoint comparators to facilitate re-entry into the internal environment.
3. The Code Function program sets up the re-entry jump address in order to gain control after the breakpoint occurs.
4. The Code Function program transfers control to the program copied to the target system.
5. After the target program begins executing, the operator must depress the RUN BKPT Key to arm the interrupt system. The target program may now give control back to the internal program any time it wishes by accessing the breakpoint address.
6. When the breakpoint occurs, the internal program may read results left in RAM by the target system routine and take whatever additional action is desired.

8-8 CODE FUNCTION EXAMPLES

Two examples of Code Function programs are given in this section. The first example is a very simple routine that writes a range of target system memory to zeros.

EXAMPLE 1:

```
                ORG 2000H
                JMP CODE0         ;CODE 0 ENTRY
                JMP XECUTE        ;CODE 1
                JMP XECUTE        ;CODE 2
                JMP XECUTE        ;CODE 3
                JMP XECUTE        ;CODE 4
                JMP XECUTE        ;CODE 5
                JMP XECUTE        ;CODE 6
                JMP XECUTE        ;CODE 7
                JMP XECUTE        ;CODE 8
                JMP XECUTE        ;CODE 9
;
;INITIALIZE—
CODE0           LXI H, 6000H      ;INITIALIZE MEMORY POINTER
                MVI B, 00H        ;INITIALIZE BYTE COUNT
;
;NOW LOOP TO CLEAR EACH TARGET MEMORY
;LOCATION FROM 6000H to 60FFH
C1              SUB A             ;CLEAR ACCUMULATOR
                CALL STM          ;STORE TO TARGET SYSTEM
                INX H             ;INCREMENT TO NEXT ADDRESS
                DCR B             ;DECREMENT BYTE COUNT
                JNZ C1            ;LOOP UNTIL COUNT EQUALS ZERO
;
;EXIT TO CONTROL PROGRAM
                JMP XECUTE
;
;DEFINE SUBROUTINE ADDRESSES
STM             EQU 1000H
XECUTE          EQU 1025H
                END
```

This example illustrates the following points:

1. The program originates at location 2000_{16} because this is the start of the address range allocated for the Diagnostic PROM.
2. The first instruction tells the program to jump to the actual starting point of the CODE 0 program. This jump instruction provides room for the other entry points, each having its own jump instruction. In this simple example, only one Code Function is implemented; consequently a full set of jump instructions, as shown, is really not needed. Notice the CODE 0 entry point has a jump instruction to the program; all other entry points jump to a routine labeled XECUTE. The XECUTE routine is one of several ways to exit a Code Function, giving control back to the Diagnostic Emulator firmware.
3. The Code Function program is written in standard 8085/8080 assembly language.

4. Whenever the Code Function program wishes to access the target system memory space, it may most easily do so by using subroutines already present in the Diagnostic Emulator firmware. In this example, the STM subroutine is called to perform the write operation to the target system.
5. When the Code Function program has finished executing, it returns control to the Diagnostic Emulator firmware by jumping to the XECUTE routine.
6. The EQU statements inform the assembler of the addresses of routines within the Diagnostic Emulator firmware—in this case, the addresses of the STM and XECUTE entry points.

The second example is a 'scope loop program. This program rotates a bit through all eight positions of an output port. The port address used is selected as is usual for other built-in functions. The routine outputs a synchronizing pulse to the BKPT-A output pin of the auxiliary connector just before rotating the bit and outputs a pulse to the BKPT B output pin after it has moved the bit through all eight positions of the output port. This program loops indefinitely and must be terminated with the RESET Key.

EXAMPLE 2:

```
               ORG 2000H
               JMP CODE0
CODE 0         LXI SP, 03060H          ;SET UP STACK
;
;OUTPUT START PULSE TO BKPT A OUTPUT—
C1             MVI A, 0FFH
               STA 034C0H              ;TURN BKPT A ON
               SUB A
               STA 034C0H              ;TURN BKPT A OFF
;
;ROTATE BIT THROUGH OUTPUT PORT
               LDA 030FDH              ;PICK UP PORT ADDRESS
               MOV C, A
               MVI A, 01H              ;SET ACCUMULATOR BIT
C2             CALL STIO               ;OUTPUT TO TARGET PORT
               ADD A                   ;SHIFT BIT LEFT
               JNZ C2
;
;FLOW HERE WHEN THE BIT HAS SHIFTED OUT
;OF THE ACCUMULATOR. NOW OUTPUT THE STOP
;PULSE TO THE BKPT B OUTPUT.
               MVI A, 0FFH
               STA 034C4H              ;TURN BKPT B ON
               SUB A
               STA 034C4H              ;TURN BKPT B OFF
;
;NOW REPEAT
               JMP C1
;
;DEFINE SUBROUTINE ADDRESS
STIO           EQU 01006H
               END
```

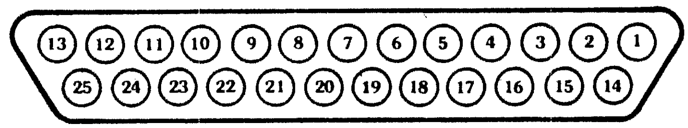
SECTION 9

SUPPLEMENTARY INFORMATION

9-1 AUXILIARY CONNECTOR

The EM-188 Diagnostic Emulator has a back panel auxiliary connector that is used as the connection point for the RS-232C communications signals and various other signals. Figure 9-1.1 shows the pinout of the auxiliary connector and the signals present on each of the pins.

Figure 9-1.1. J3-Auxiliary Connector Pinout (D-Subminiature, Female)



Pin		Pin	
1	PROTECTIVE GROUND	14	
2	SERIAL DATA (OUT) (RS-232C)	15	
3	SERIAL DATA (IN) (RS-232C)	16	
4	REQUEST-TO-SEND (OUT) (RS-232C)	17	
5	CLEAR-TO-SEND (IN) (RS-232C)	18	
6	DATA-SET-Ready (NOT IMPLEMENTED)	19	
7	SIGNAL GROUND (RS-232C)	20	DATA TERMINAL READY (OUT) (RS-232C)
8			
9		21	
10	EXT BREAK (IN)	22	$\overline{\text{RUN}}$ (OUT)
11	TRACE HOLD (IN)	23	+5 VOLTS (OUT)
12	BKPT A and SA START (OUT)	24	GROUND
13	BKPT B and SA STOP (OUT)	25	SIGNATURE CLOCK (OUT)

Pins without associated signals shown are not connected within the EM-188.

The functions of the auxiliary connector signals are summarized below:

Pin 1 Protective Ground: Connected in the EM-188 to the chassis, and from the chassis to the protective ground terminal of the primary power input connector.

Pin 2 Serial Data Out: This signal is driven to nominal ± 12 volt levels by an RS-232C compatible driver. See Section 9-3 (Serial Interface) for format of serial data.

Pin 3 Serial Data In: The EM-188 accepts data on this pin that has voltage levels as specified by the EIA RS-232C specification and the format given in Section 9-3.

Pin 4 Request to Send: This signal is driven to nominal ± 12 volt levels by an RS-232C compatible driver. The state of this signal is determined by software in the EM-188.

Pin 5 Clear to Send: The EM-188 accepts a signal on this pin having RS-232C voltage levels. The state of this signal may be read by the EM-188 control software.

Pin 7 Ground: Connected in the EM-188 to the system logic ground which is isolated from the protective ground (Pin 1). Note, however, that this ground is connected to the emulator probe ground pin; then when the EM-188 is connected to the target equipment, the target system logic ground and the EM-188 logic ground are connected together and to the ground system of the equipment plugged into the Auxiliary Connector.

Pin 10 External break (In): A TTL level input with an internal 3.3K pull-up resistor. If this input is pulled low, the Diagnostic Emulator stops executing the target program as though STEP were depressed or an Internal Breakpoint were detected. (If the Diagnostic Emulator is already in PAUSE, this has no effect.) This input stops execution even when the breakpoints are not enabled.

Pin 11 Trace Hold (In): A TTL level input with an internal 3.3K pull-up resistor. If the Diagnostic Emulator is executing a target program and this input is pulled low, further updating of the Trace Memory stops, although the program continues to execute. The contents of the Trace Memory are effectively frozen, and can be reviewed later after program execution has been halted.

Pin 12 BKPT A and SA START (Out): A TTL level output providing a high-going pulse at the time breakpoint conditions are satisfied for the Breakpoint A Comparator. This signal can be used to trigger an oscilloscope at a particular point of program execution. It can also be used as the START signal for a signature analyzer. This signal may be set high or low under software control when the Diagnostic Emulator is in PAUSE. This permits diagnostic routines to generate sync pulses or signature analyzer START signals under direct program control.

Pin 13 BKPT B and SA STOP (Out): A TTL level output associated with the Breakpoint B Comparator. It is functionally identical with the BKPT A signal described above.

Pin 20 DATA TERMINAL READY (Out): This signal is driven to a nominal +12 volts to indicate that the EM-188 is ready to send data. Its signal state does not change.

Pin 22 RUN (Out): A TTL level output that is active (low) if the EM-188 is executing the target program or accessing the target address space.

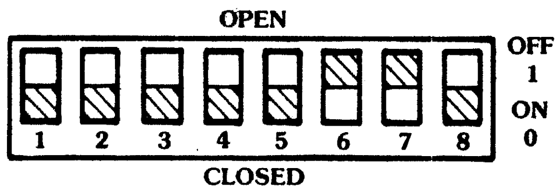
Pin 23 +5 VOLTS (Out): 0.5 amp max.

Pin 24 GROUND: +5 volt return line. Internally connected to signal ground (pin 7).

Pin 25 SIGNATURE CLOCK (Out): A buffered RD signal from the CPU. It is primarily used as a clock for signature analysis testing of equipment for which the EM-188 provides the stimulus.

9-2 OPTION SWITCHES

The EM-188 has a set of eight small switches that are accessible from the back panel of the machine. These switches are used to select optional operating characteristics of the EM-188 reset circuitry and communications interface.



SWITCH:

1 Not used.

2 Not used.

3 If CLOSED, EM-188 ignores clear-to-send (CTS) signal and communications software will output data at any time on operator command. If OPEN, EM-188 will output data only if clear-to-send is in the ON (positive) state.

4 If CLOSED, target system RESET signal will reset the EM-188 in the same manner as the RESET Key. If OPEN, target system RESET signal will reset EM-188 emulation CPU but the operator station will not be reset. This makes it possible to emulate systems in which the CPU is made to restart at intervals as part of the normal operation of the system.

5 If CLOSED, EM-188 RESET signal (from RESET Key or power-on-reset) is sent to target system RESET through the CPU reset pin. If OPEN, no reset of the target system is attempted.

6,7,8 Set up characteristics of serial communications interface as shown in Table 9-2.1.*

Table 9-2.1. Set-up Characteristics. for Serial Port

SW6	SW7	SW8	DATA BITS PER CHARACTER	STOP BITS	
0	0	0	5 **	1	
0	0	1	5 **	1½	
1	0	0	6 **	1	
1	0	1	6 **	2	
0	1	0	7	1	
0	1	1	7	2	
1	1	0	8	1	Normal Set-up
1	1	1	8	2	

* *CLOSED 0, OPEN 1*

** *Standard EM-188 communications software requires at least 7 bits for operation.*

9-3 SERIAL INTERFACE

The EM-188 Serial Interface is compatible with the RS-232C standard pin conventions and signaling levels. The signals and connections are given in Section 9-1 (Auxiliary Connector).

The format of a serial word is shown in Figure 9-3.1. When no data is being transmitted, the Serial Data Out pin will be at the -12 volt level (marking). When the EM-188 sends a character, there will always be a START bit, followed by 5, 6, 7 or 8 DATA bits, and 1, 1.5 or 2 STOP bits. The number of DATA bits and STOP bits are selected by the Option Switches on the back panel. See Section 9-2 (Option Switches).

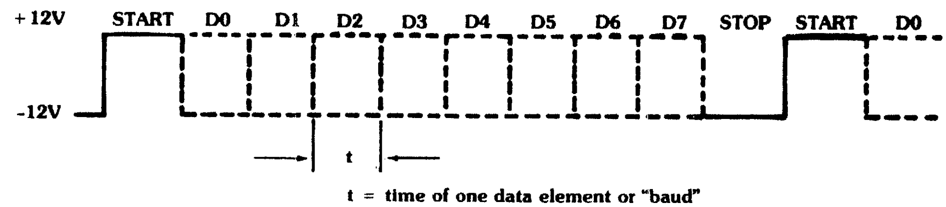
The standard EM-188 software transmits and receives ASCII characters which require 7 bits for their representation. For this reason, the option switches must be set for 7 or 8 bit characters for proper operation. Some data terminals require two stop bits for proper operation and the EM-188 will operate with these terminals; one stop bit is recommended for most other terminals because a somewhat higher data rate is obtained if time is not given to unneeded stop bits.

The EM-188 with standard software does not send or check parity. However, it is possible to have one of the data bits function as a parity bit if the parity generation and checking is done by software.

Two additional signals that are used by the EM-188 are the Request-to-Send (Pin 4) output and the Clear-to-Send (Pin 5) input. The EM-188 standard software uses these signals to coordinate the data transfer. When the EM-188 is ready to begin receiving data, it changes the Request-to-Send line from low to high and awaits data transmission. When the EM-188 has finished receiving data, it will return the Request-to-Send line to the low state. When the EM-188 is ready to send a character, the software tests the condition of the Clear-to-Send line and transmission of the character proceeds only if Clear-to-Send is in the high state; the character is held if the signal is in the low state. Thus, a receiving device may control the transfer of data by taking the Clear-to-Send line high when more data is desired and low when not ready for data. The EM-188 may be made to consider the Clear-to-Send line as always high by closing Option Switch 3 on the back panel.

The serial port transmission rate is controlled by the rotary hexadecimal switch in the lower left corner of the back panel. The EM-188 is capable of communicating at baud rates from 50 Baud to 19,200 Baud. See Figure 9-3.1.

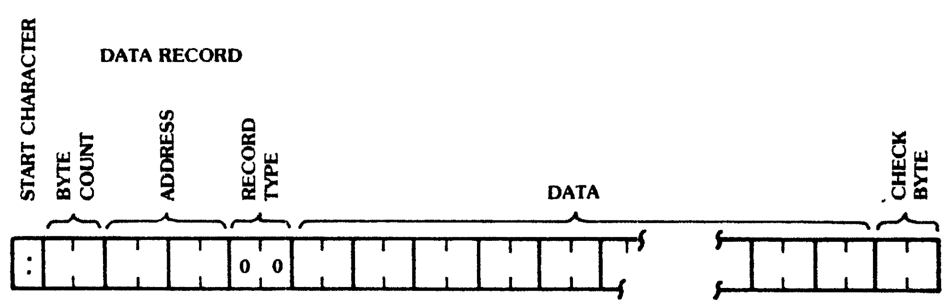
Figure 9-3.1. Serial Word Format.



Switch Position	Baud Rate	t	
D	50	20	mSEC
C	75	13.33	
0	110	9.09	
B	134.5	7.43	
1	150	6.67	
A	200	5	
2	300	3.33	
9	600	1.67	
4	1,200	833	uSEC
5	1,800	556	
3, 8	2,400	417	
6	4,800	208	
7	9,600	104	
E, F	19,200	52	

9-4 UPLOAD/DOWNLOAD PROTOCOL

The EM-188 routines CODE C3 and CODE C4 initiate routines to load the target memory space with data from the serial link or dump data from the target address space to the serial link. The EM-188 uses a particular format to transfer the data. This format is compatible with the Intel family of development systems.



START CHARACTER
An ASCII colon is used to signal the start of a record.

BYTE COUNT
Two ASCII characters representing hexadecimal digits giving the number of data bytes in the record.

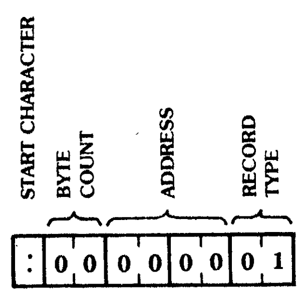
ADDRESS
Four ASCII characters representing hexadecimal digits giving the address in target memory where the first of the data bytes of this record is to be located. The following bytes in the record are located in sequentially higher addresses in memory.

RECORD TYPE
Two ASCII characters representing hexadecimal digits that are encoded to designate the record type. For data records as defined here the record type will always be 00_{16}.

DATA
Each two ASCII characters representing hexadecimal digits give the bit pattern of one eight-bit byte of data. The total number of data bytes in the record is given by the byte count.

CHECK BYTE
Two ASCII characters representing hexadecimal digits giving the value of a check byte. The value of the check byte is the two's complement of the sum of all the other bytes of the record; that is, the byte count plus the first byte of the address plus the second byte of the address plus the record type byte plus all of the data bytes. If the check byte is added to the sum of all the other bytes, the result will be zero. (The addition is performed modulo 256 in that any carry out of an eight bit result is ignored.)

END-OF-FILE RECORD



START CHARACTER
An ASCII colon is used to signal the start of a record.

BYTE COUNT
Two ASCII zeros.

ADDRESS FIELD
Four ASCII charaters representing hexadecimal zeros, or the starting address of the program.

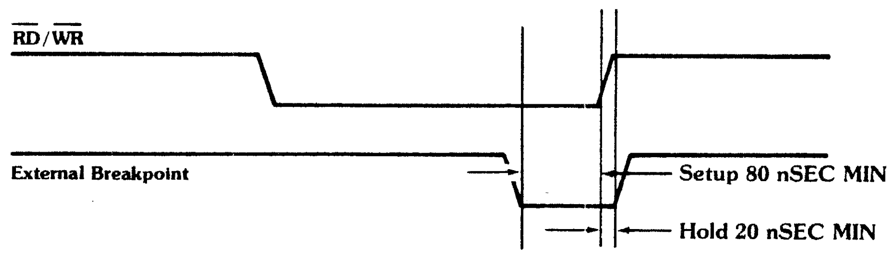
RECORD TYPE FIELD
Two ASCII characters representing the hexadecimal digits 01.

9-5 EXTERNAL BREAKPOINT

The EM-188 Diagnostic Emulator is provided with an input that permits an external signal to halt the execution of the target program when the EM-188 is in the RUN mode. Pin 10 of the back panel Auxiliary Connector (J3) is the input connection. External Breakpoint is a TTL level input with a 3.3K resistor pull up to +5 volts. If this input is in the high state, or if the input is left open, then the EM-188 will run the target program in the normal manner. If this input is pulled low, the target program will halt; if the target program is already halted, the External Breakpoint signal will have no effect.

The EM-188 samples the External Breakpoint input at the trailing edge (low-to-high transition) of the RD or WR signal of the CPU. If the signal is low at the sample time, the signal is entered into the Trace Memory, thus marking the cycle during which the signal was detected; circuitry in the EM-188 is also armed to halt program execution after completion of the current instruction. When the target program has been halted, the EM-188 firmware will determine which cycle of the last instruction caused the breakpoint and the Trace Memory will be positioned to display that cycle. Figure 9-5.1 shows the timing relationships of the External Breakpoint signal.

Figure 9-5.1. Timing Relationships.



SETUP PRIOR TO LOW-HIGH TRANSITION OF $\overline{RD}$ OR $\overline{WR}$: 80nSEC MINIMUM.
HOLD TIME AFTER LOW-HIGH TRANSITION OF $\overline{RD}$ OR $\overline{WR}$: 20nSEC MINIMUM.

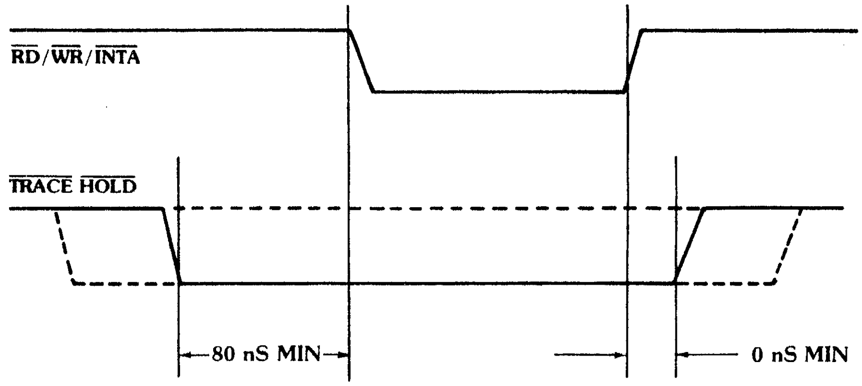
9-6 TRACE HOLD

The EM-188 Diagnostic Emulator is provided with an input that permits external equipment to control the tracing of program execution. Pin 11 of the back panel Auxiliary Connector (J3) is the Trace Hold input. Trace Hold is a TTL level input with a 3.3K resistor pull up to +5 volts. If this input is in the high state, or if the input is left open, then the Trace Memory operates normally. If this input is pulled low, the Trace Memory stops tracing program execution.

The circuitry controlling the Trace Hold input must ensure that set-up and hold time requirements are met for reliable operation. The requirements are shown in Figure 9-6.1.

The Trace Hold feature may be used to capture trace data on a selective basis as detailed in the following paragraphs.

Figure 9-6.1. Trace Hold and Timing

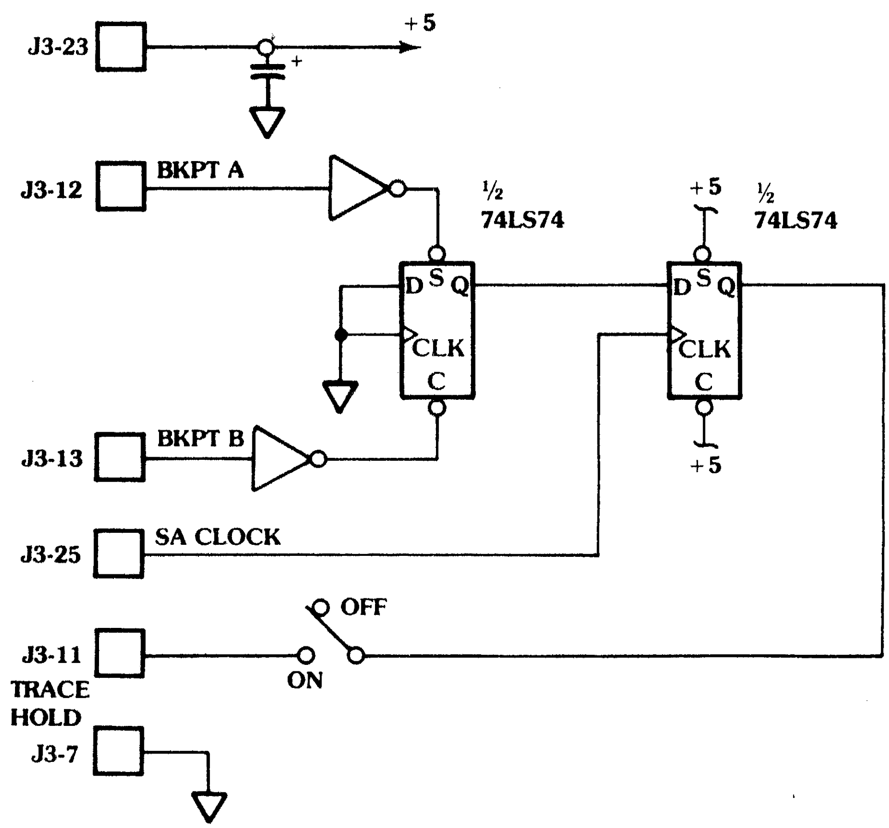


SET-UP TIME:
AT LEAST 80 nSEC BEFORE HIGH TO LOW TRANSITION OF RD, WR OR INTA.
HOLD TIME:
0 nSEC AFTER LOW-TO-HIGH TRANSITION OF RD, WR OR INTA.

9-6.1 WINDOW MODE

Figure 9-6.2 shows a schematic of a simple external circuit that may be used to implement window mode operation of the Trace Memory. This circuit controls the Trace Hold input of the EM-188 so that only bus activity that occurs after the Breakpoint A address through one cycle past the Breakpoint B address is recorded.

Figure 9-6.2. Window Mode Circuit



TRACE STARTS AT ADDR A+1 AND STOPS AT ADDR B+1.

9-6.2 SELECTIVE TRACE

External circuitry may be designed for a variety of selective tracing functions. An example is an application in which it is desired to use the capacity of the Trace Memory to capture cycles written to a particular I/O port; the I/O port select signal (port decode) may be routed to the Trace Hold input to permit Trace Memory operation only when the port select signal is active. When execution is halted and the Trace Memory contents is reviewed, only bus cycles to the I/O port will be seen.

9-7 SIGNATURE ANALYSIS

In 1977, the Hewlett Packard Company introduced a digital servicing technique called signature analysis. The signature analysis technique requires first of all that the system under test be stimulated to cause repetitive patterns or bit streams to occur at various circuit nodes of the system. When such a stimulus is applied, it is possible to use an instrument such as the HP 5004A Signature Analyzer to observe these bit streams and convert them to four-digit hexadecimal displays on the front panel of the instrument. The bit stream, or pattern of lows and highs present at a given point in a circuit, is the "signature" of that circuit node. Faulty components, opens, shorts and other circuit defects will almost always cause alteration of a signature that may be observed by the signature analyzer.

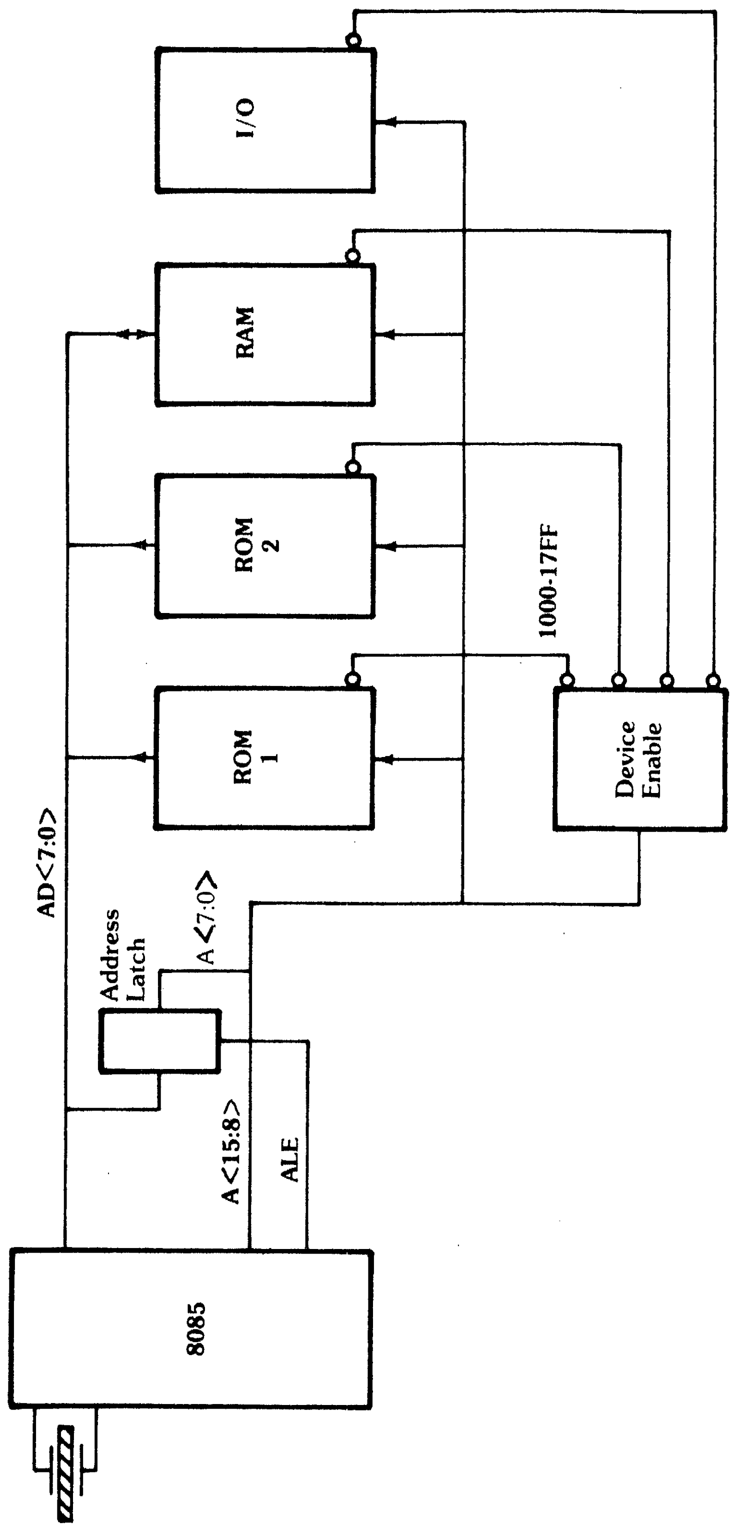
The EM-188 Diagnostic Emulator does not contain circuitry for examining signatures at circuit nodes. It does, however, contain pre-programmed stimulus routines that may be used to generate the repetitive signals that must be present for the signature analysis concept to work.

Figure 9-7.1 is a simplified microprocessor system diagram and shows an 8085 processor with its address latch, two ROMs, one RAM, some I/O circuitry and device enable logic. To test a system such as this one, first perform the obvious checks such as measurement of the supply voltages and then connect the EM-188 to the circuit. The system clock may be checked by using the CODE D2 function; the clock frequency displayed by the EM-188 should be one-half of the crystal frequency. Now proceed with signature analysis testing by connecting a signature analyzer (such as the HP 5004A) to the EM-188 Auxiliary Connector (J3, on the back panel of the EM-188) as follows:

SA GROUND	to J3 - 24	(GROUND)
SA START	to J3 - 12	(BKPT A and SA START)
SA STOP	to J3 - 13	(BKPT B and SA STOP)
SA CLOCK	to J3 - 25	(SIGNATURE CLOCK)

The selector switches for the START, STOP and CLOCK signals on the signature analyzer should be set for low-to-high edge recognition (buttons out on the HP 5004A).

Figure 9.7-1 Simplified microprocessor system diagram.



BINARY ADDRESS TEST (FREE-RUN)

The EM-188 Diagnostic Emulator has a built-in mode which places the CPU in a free-run condition by inserting continuous NOP instructions into the CPU. As a result, the processor outputs successive addresses, along with the RD signal; this mode is useful as a signature analysis stimulus routine because it stimulates all of the address lines in the system and since the RD signal is active, any devices in an operating system will drive the data bus when appropriate addresses are present.

To start with, set both the A and B breakpoint comparators so that they will respond to READ cycles at address 0000_{16} (See Section 4-1.4). Next, start the binary address routine by depressing the keys for CODE B5. The EM-188 will begin to output incrementing addresses; a SA START pulse and an SA STOP pulse will occur each time address 0000_{16} is output. The stimulus and signature analyzer are now ready for use.

At this point, the address bus signals may be probed with the signature analyzer and each should display its characteristic signature. The various device enable signals may be probed and, if the system circuitry is working correctly, characteristic signatures will be obtained. Various nodes internal to the Device Enable Logic may also be probed; in short, any circuit point may be tested where the signal present is determined by the address inputs and the RD signal.

In most cases, the data bus cannot be tested with this setup because the data bus signals are not determined by addresses for all possible address values. For example, some addresses may result in floating the data bus; other addresses may select RAMs whose contents are not known. Therefore, to test the data bus using the signature analysis technique, it is necessary to restrict the start-stop window of the signature analyzer so that the data bus is sampled only when addresses are present that should result in known data on the bus.

Suppose that it is known that ROM 1, in Figure 9-7.1, is enabled by the Device Enable Logic for any address in the range from 1000_{16} to $17FF_{16}$. If the SA START signal could be generated when the incrementing address reaches 1000_{16} and the SA STOP signal generated when the incrementing address reaches $17FF_{16}$, then signatures would be computed only during the time the data bus contained deterministic data. The SA START and SA STOP signals may be easily adjusted to occur at any desired addresses by setting the appropriate breakpoint values into the A and B breakpoint comparators. For the example just given, set the A comparator to respond to READ cycles at address 1000_{16} and set the B comparator to respond to READ cycles at address $17FF_{16}$. Then test the eight data lines to obtain the characteristic signatures. Note that the signatures obtained depend not only on the details of the circuitry of the system under test, but also on the contents of the ROM involved; consequently, this test also verifies that the ROM contains the same pattern as the ROM for which the reference signatures were originally obtained.

The EM-188 also has a built-in test function for obtaining a signature of a ROM in a system, and no signature analyzer is needed. The test is set up by entering the first and last address of the ROM into the BEG and END registers of the EM-188 to define the range over which the routine will operate. Then start the routine with the Keys for CODE D3. The routine will execute and then display a four-digit hexadecimal signature on the EM-188 front panel. The signature obtained does not have any simple relationship to signatures obtained with the HP 5004A; for one thing, the CODE D3 algorithm operates on all eight data bits of the ROM word simultaneously while the eight signatures obtained by the HP 5004A for a ROM are computed from one "bit slice" of the ROM at a time. In addition, the generating polynomial used by the EM-188 routine differs from that used by the HP 5004A. See Section 7-4 for additional information.

Other routines that are programmed in the EM-188 Diagnostic Emulator may be useful as stimulus routines for signature analysis testing. The CODE B6 routine repetitively stores a data pattern and the complement of that pattern to a selected address. An I/O port, such as the one shown in Figure 9-7.1, may be tested by storing the complementing data to the I/O port and observing the signatures obtained at the output side of the I/O port. In special cases it may be found necessary to write custom CODE function routines to stimulate a system in a way that useful signatures may be obtained. As an example, consider the problem of obtaining a signature at the outputs of an LSI interface chip such as the Intel 8255A Programmable Peripheral Interface. This device requires that various control registers and data direction bits be set up for the intended application before data transfers are performed. A custom CODE Function routine can easily perform the desired set-up and then generate the stimulus for signature analyzer probing.

For additional information on Signature Analysis testing, see the following publications:

1. "Hexadecimal Signatures Identify Troublespots in Microprocessor Systems", Gary Gordon and Hans Nadig. ELECTRONICS, March 3, 1977.
2. Application Note 222, "A Designer's Guide to Signature Analysis", Hewlett Packard Corporation.

9-8 SOFT SHUTDOWN

In some applications it is desirable to halt emulation of the target program when a particular event occurs or when a particular address is reached; after this, it is necessary that the processor execute a program to shut down the target system equipment in an orderly manner. For example, there may be hammer driver coils which would burn up if left energized. The EM-188 may be configured to operate in systems with requirements like these by writing the needed soft shutdown routines and programming them into an EPROM that is then plugged into the Diagnostic PROM socket. A small Code Function program is also required to insert the re-entry jump instruction into the EM-188 internal scratchpad RAM. When enabled, the soft shutdown routine would be executed every time the emulator transfers from RUN to PAUSE, and also after each single-step instruction execution. After executing, the soft-shutdown subroutine should exit to the monitor routine with a return instruction.